

Bijaya Ketan Panigrahi
Ponnuthurai Nagarathnam Suganthan
Swagatam Das
Shubhransu Sekhar Dash (Eds.)

LNCS 8297

Swarm, Evolutionary, and Memetic Computing

4th International Conference, SEMCCO 2013
Chennai, India, December 2013
Proceedings, Part I

1
Part I

 Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Bijaya Ketan Panigrahi
Ponnuthurai Nagarathnam Suganthan
Swagatam Das
Shubhransu Sekhar Dash (Eds.)

Swarm, Evolutionary, and Memetic Computing

4th International Conference, SEMCCO 2013
Chennai, India, December 19-21, 2013
Proceedings, Part I

Volume Editors

Bijaya Ketan Panigrahi
IIT Delhi, New Delhi, India
E-mail: bijayaketan.panigrahi@gmail.com

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

Swagatam Das
Indian Statistical Institute, Kolkata, India
E-mail: swagatamdas19@yahoo.co.in

Shubhransu Sekhar Dash
SRM University, Tamil Nadu, India
E-mail: munu_dash_2k@yahoo.com

ISSN 0302-9743

e-ISSN 1611-3349

ISBN 978-3-319-03752-3

e-ISBN 978-3-319-03753-0

DOI 10.1007/978-3-319-03753-0

Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013954557

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

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

© Springer International Publishing Switzerland 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, 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.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

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 4th International Conference on Swarm, Evolutionary and Memetic Computing (SEMCCO 2013) held during December 19–21, 2013, at SRM University, Chennai, 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 progresses in the cutting-edge research on swarm, evolutionary, memetic and other computing techniques such as neural and fuzzy computing, to explore new application areas, to design new nature-inspired algorithms for solving hard problems, and finally to create awareness about these domains to a wider audience of practitioners.

SEMCCO 2013 received 350 paper submissions from 20 countries across the globe. After a rigorous peer-review process involving 1,100 reviews, 126 full-length articles were accepted for oral presentation at the conference. This corresponds to an acceptance rate of 36% and is intended to maintain the high standards of the conference proceedings. The papers included in this LNCS volume cover a wide range of topics in swarm, evolutionary, memetic, fuzzy, and neural computing algorithms and their real-world applications in problems from diverse domains of science and engineering.

The conference featured distinguished keynote speakers: Prof. Marios M. Polycarpou, President, IEEE Computational Intelligence Society and Director, KIOS Research Center for Intelligent Systems and Networks Department of Electrical and Computer Engineering, University of Cyprus; Prof. Ferrante Neri, Professor of Computational Intelligence Optimization, De Montfort University, UK; Dr. M. Fatih Tasgetiren, Associate Professor of Industrial Engineering, Yasar University, Turkey; Dr. Dipti Srinivasan, Associate Professor, Department of Electrical and Computer Engineering, National University of Singapore. The other prominent speakers were Dr. P.N. Suganthan, NTU, Singapore; Dr. Adel Nasiri, Department of Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee, USA; Dr. Ravipudi Venkata Rao, NIT, Surat, India; and Dr. Swagatam Das, ISI, Kolkata, India.

We take this opportunity to thank the authors of the submitted papers for their hard work, adherence to the deadlines, and patience with the review process. The quality of a referred volume depends mainly on the expertise and dedication of the reviewers. We are indebted to the Program Committee/Technical Committee members, who produced excellent reviews in short time frames.

We would also like to thank our sponsors for providing all the logistical support and financial assistance. First, we are indebted to SRM University Management and Administration for supporting our cause and encouraging us to organize the conference at SRM University, Chennai, India. In particular, we would like to express our heartfelt thanks for providing us with the necessary

financial support and infrastructural assistance to hold the conference. Our sincere thanks to Thiru T.R. Pachamuthu, Chancellor, Shri P. Sathyanarayanan, President, Dr. M. Ponnavaikko, Vice-Chancellor, Dr. N. Sethuraman, Registrar, and Dr. C. Muthamizchelvan, Director (E&T) of SRM University, for their encouragement and continuous support. We thank Prof. Carlos A. Coello Coello, Prof. Nikhil R. Pal, and Prof. Rajkumar Roy for providing valuable guidelines and inspiration to overcome various difficulties in the process of organizing this conference.

We would also like to thank the participants of this conference. Finally, we would like to thank all the volunteers who made great efforts in meeting the deadlines and arranging every detail to make sure that the conference could run smoothly. We hope the readers of these proceedings find the papers inspiring and enjoyable.

December 2013

Bijaya Ketan Panigrahi
Swagam Das
P.N. Suganthan
S.S. Dash

Organization

Chief Patron

Thiru T.R. Pachamuthu

Patron

Shri P. Sathyanarayanan

Honorary Chairs

Nikhil R. Pal, India
Carlos A. Coello Coello, Mexico
Rajkumar Roy, UK
M. Ponnaivaikko, India

General Chairs

B.K. Panigrahi, India
Swagatam Das, India
P.N. Suganthan, Singapore

Program Chairs

S.S. Dash, India
Zhihua Cui, China
J.C. Bansal, India

Program Co-chairs

K. Vijaya Kumar, India
A. Rathinam, India

Steering Committee Chair

P.N. Suganthan, Singapore

Publicity Chairs

S.S. Dash, India
S.C. Satpathy, India
N.C. Sahoo, Malaysia

Special Session Chairs

Sanjoy Das, USA
Wei-Chiang Hong, Taiwan

R. Rajendran, India
E. Poovammal, India

Tutorial Chair

S.K. Udgata, India

Technical Program Committee

Abbas Khosravi
Ahmed Y. Saber

Deakin University, Australia
Senior Power System Engineer - R&D
Department, OTI/ETAP, CA, USA

Aimin Zhou
Almoataz Youssef Abdelaziz
Athanasios V. Vasilakos
Ayman Abd El-Saleh
Balasubramaniam Jayaram

East China Normal University, China
Ain Shams University, Cairo, Egypt
University of Western Macedonia, Greece
Multimedia University, Cyberjaya, Malaysia
Indian Institute of Technology Hyderabad,
India

Carlos A. Coello Coello
Chilukuri K. Mohan
Chanan Singh
Dipankar Dasgupta
Dinesh Kant Kumar
Esperanza García-Gonzalo
Ganapati Panda
G. Kumar Venayagamoorthy
G.A. Vijayalakshmi Pai
Gerardo Beni
Halina Kwasnicka
Hisao Ishibuchi
Hong-Jie Xing
Janusz Kacprzyk
John MacIntyre
Jeng-Shyang Pan

CINVESTAV-IPN, México
Syracuse University, USA
Texas A&M University, USA
University of Memphis, USA
RMIT, Australia
Oviedo University, Spain
IIT Bhubaneswar, India
Clemson University, USA
PSG College of Technology, Coimbatore, India
University of California, Riverside, USA
Wroclaw University of Technology, Poland
Japan
Hebei University, China
Systems Research Institute, Poland
University of Sunderland, UK
National Kaohsiung University of Applied
Sciences, Taiwan

Juan Luis Fernández Martínez
Kalyanmoy Deb
Konstantinos E. Parsopoulos
K. Vaisakh
Laxmidhar Behera
Leandro Nunes de Castro
Lingfeng Wang

Universidad de Oviedo, Spain
IIT Kanpur, India
University of Ioannina, Greece
Andhra University, India
IIT Kanpur, India
Universidade Presbiteriana Mackenzie, Brazil
The University of Toledo, USA

| | |
|---------------------------|---|
| M.A. Abiso | King Fahd University of Petroleum & Minerals, Saudi Arabia |
| Maurice Clerc | France Telecom R&D, France |
| Manoj Kumar Tiwari | IIT Kharagpur, India |
| Martin Middendorf | University of Leipzig, Germany |
| Meng Hiot Lim | NTU, Singapore |
| N.C. Sahoo | IIT Bhubaneswar, India |
| Oscar Castillo | Tijuana Inst. Technology, Mexico |
| Peng Shi | University of Adelaide, Australia |
| Pei Chan Chang | Taiwan |
| P.K. Dash | SOA University, India |
| Quan Min Zhu | University of the West of England, Bristol, UK |
| Rafael Stubs Parpinelli | State University of Santa Catarina, Brazil |
| Saeid Nahavandi | Deakin University, Australia |
| Samrat Sabat | University of Hyderabad, Hyderabad, India |
| Satchidananda Dehuri | Ajou University, South Korea |
| Shu-Heng Chen | National Chengchi University, Taipei, Taiwan, ROC |
| S.G. Ponnambalam | Monash University, Malaysia |
| Siba K. Udgata | University of Hyderabad, Hyderabad, India |
| Saman K. Halgamuge | Australia |
| Sanjoy Das | Kansas State University, USA |
| S. Baskar | Thiagarajar College of Engineering, India |
| Somanath Majhi | IIT, Guwahati, India |
| Tan Kay Chen | National University of Singapore, Singapore |
| Vincenzo Piuri | Università degli Studi di Milano, Italy |
| V. Ravi | IDRBT, Hyderabad, India |
| Wei-Chiang Hong | Oriental Institute of Technology, Taiwan |
| Xin-She Yang | Middlesex University, London, UK |
| X.Z. Gao | Aalto University, Finland |
| Yew-Soon Ong | Nanyang Technological University, Singapore |
| Yuehui Chen | University of Jinan, China |
| Yuhui Shi | Xi'an Jiaotong-Liverpool University, China |
| Yucheng Dong | Sichuan University, China |
| Zhao Xu | Hong Kong Polytechnic University, Hong Kong |
| Zong Woo Geem | Gachon University, South Korea |
| S. Baskar | Thiagarajar College of Engineering, Madurai, India |
| P. Somsundaram | Anna University, Chennai, India |
| D. Devaraj | Kalasalingam University, India |
| C. Christopher Asir Rajan | Pondicherry University, India |

Technical Review Board

| | | |
|-----------------------------|----------------------------------|-----------------------------|
| Abdelaziz, Almoataz | Delhibabu, K. | Kazimipoor, Borhan |
| Abhyankar, Abhijit | Dewan, Hrishikesh | Khadkikar, Vinod |
| Agarwal, Vineeta | Dhingra, Atul | Lakshmi, B. Geetha |
| Alizadeh Bidgoli, Mohsen | Elgammal, Adel | Lim, Meng-Hiot |
| Amali, Miruna Joe | Fernández Martínez, Juan Luis | Lohokare, Mohan |
| Anand, Ashish | Fu, Wenlong | Lopes, Heitor Silverio |
| Asafuddoula, Md. | Gaddam, Mallesham | Li, Miqing |
| Abido, Mohammad | Garcia Gonzalo, Esperanza | M.P., Selvan |
| Ahuja, Ashish | Geem, Zong Woo | Maity, Dipankar |
| Ansari, Abdul Quaiyum | Ghosh, Arnob | Martinovic, Goran |
| B., Chitti Babu | Ghoshal, Saktiprasad | Martinovic, Jan |
| Bakwad, Kamalakar | Grosan, Crina | Meher, Saroj K. |
| Banakara, Basavaraja | G. Sridhar Reddy, | Majhi, Babita |
| Bansal, Jagdish Chand | Gandhi, Tapan | Majhi, Ritanjali |
| Benítez, José M. | Gao, Xiao-Zhi | Mandal, Durbadal |
| Bhuvaneswari, G. | Giri, Ritwik | Mandal, Kamal |
| Bijwe, P.R. | Godarzi, Hosin | Matousek, Radek |
| Bajo, Javier | Gottipati, Pavani | Maurice, Clerc |
| Bhat, Ashoka | Gross, Roderich | Mehrotra, Kishan |
| Brest, Janez | Hassanien, Aboul Ella | Mei, Yi |
| Casola, Valentina | Hamza, Noha | Mishra, Mahesh Kumar |
| Chen, Jianyong | Hasan, Shazia | Misra, Bijan |
| Cernea, Ana | Hong, Tzung-Pei | Misra, Rakesh K. |
| Ch., Sudheer | Hong, Wei-Chiang | Mohan, Chilukuri |
| Chatterjee, Saibal | Hota, Ashish R. | Mohanty, Mihir |
| Chaturvedi, D.K. | Iacca, Giovanni | Mohanty, Saraju |
| Chawla, Akshay | Iqbal, Muhammad | Mohapatra, Ankita |
| Cui, Zhiwei | Janakiraman | Morgan, Rachael |
| Dahal, Keshav | Jayavelu, Senthilnath | Muda, Azah |
| Das, Sanjoy | Kaabouch, Naima | Muhammad Ishaque, Kashif |
| Das, Swagatam | Kai-Zhou, Gao | Mukherjee, V. |
| Dash, S.S. | Karabulut, Korhan | Murthy, J.V.R |
| Datta, Dilip | Konar, Amit | Nguyen, Su |
| De, Dipankar | Kominkova Oplatkova, Zuzana | Nanda, Pradipta Kumar |
| Dwivedi, Sanjeet | Kononova, Anna | Otjacques, Benoit |
| Dash, P.K | Kratky, Michal | P., Jayaprakash |
| Datta, Rituparna | Kwintiana, Bernadetta | Panda, Rutuparna |
| Dauwels, Justin | K.R., Krishnanand | Parida, Sanjoy |
| Davendra, Donald | Kale, Vijay | Platos, Jan |
| De, Mala | Kar, Rajib | Pati, Soumen Kumar |
| Debchoudhury, S. | | Patnaik, Amalendu |
| Dehuri, Satchidananda | | Ponnambalam, S.G. |

| | | |
|-----------------------|--------------------------|--------------------------|
| Pratihari, Dilip | Selvakumar, A. | Swain, Akshaya |
| P.P., Rajeevan | Immanuel | Thangaraj, Radha |
| Panda, Sidhartha | Senroy, Nilanjan | Thomas, Mini |
| Panda, Ganapati | Shariatmadar, Seyed | Tiwari, Manoj |
| Pandit, Manjaree | Mohammad | Tang, Ke |
| Panigrahi, Siba | Sharma, Shailendra | Thelukuntla, Chandra |
| Parillo, Fernando | Shukla, Anupam | Shekar |
| Pant, Millie | Singh, Madhu | Torkaman, Hossein |
| Pattnaik, Shyam | Singh, Mukhtiar | Udgata, S.K. |
| Pluhacek, Michal | Singh, Pramod Kumar | V., Ravikumarpandi |
| Puhan, Niladri | Swarup, Shanti | Vaisakh, K. |
| Qian, Bin | Sabat, Samrat | Verma, Nishchal |
| Rao, Ravipudi | Salkuti, Surender Reddy | Vijay, Ritui |
| Rocky, Taif Hossain | Samantaray, Subhansu | Wang, Lingfeng |
| Rout, Pravat Kumar | Satapathy, Suresh | Wang, Shengyao |
| Rybnik, Mariusz | Chandra | Willjuice |
| Rajagopal, V. | Schaefer, Gerald | Iruthayarajan, M. |
| Rahman, Humyun Fuad | Senkerik, Roman | Xie, Liping |
| Rajasekhar, Anguluri | Sethuraman, Kowsalya | Yang, Xin-She |
| Ramasamy, Savitha | Shieh, Chin-Shiuh | Yusof, Norazah |
| Ravi, V. | Shrivastava, Ashish | Yusof, Salman |
| Ravishankar, Jayashri | Shrivastava, Nitin Anand | Zamuda, Ales |
| Ren, Ye | Singh, Bhim | Zapotecas Martínez, Saúl |
| | Singh, Manohar | Xie, Feng |
| S., Sudha | Singh, Sanjeev | Xue, Bing |
| Saikia, Lalit | Sinha, Dr. Nidul | Zelinka, Ivan |
| Salehinejad, Hojjat | | Zhiehua, Cui |
| Samiei Moghaddam, | Sishaj P. Simon | Zhou, Aimin |
| Mahmoud | Skanderová, Lenka | Zhuhadar, Leyla |
| Saxena, Anmol Ratna | Sun, Jianyong | |

Organizing/Technical Program Committee

R. Jegatheesan, SRM University
 R. Ramanujam, SRM University
 N. Chellammal, SRM University
 C.S. Boopathy, SRM University
 D. Suchithra, SRM University
 K. Mohanraj, SRM University
 N. Kalaiarasi, SRM University
 R. Sridhar, SRM University
 D. Sattianandan, SRM University
 C. Bharathiraja, SRM University
 S. Vidyasagar, SRM University
 C. Subramanian, SRM University

Table of Contents – Part I

| | |
|---|----|
| A Populated Iterated Greedy Algorithm with Inver-Over Operator for Traveling Salesman Problem | 1 |
| <i>M. Fatih Tasgetiren, Ozge Buyukdagli, Damla Kiziaz, and Korhan Karabulut</i> | |
| Meta-modeling and Optimization for Varying Dimensional Search Space | 13 |
| <i>Kalyanmoy Deb, Soumil Srivastava, and Akshay Chawla</i> | |
| A General Variable Neighborhood Search Algorithm for the No-Idle Permutation Flowshop Scheduling Problem | 24 |
| <i>M. Fatih Tasgetiren, Ozge Buyukdagli, Quan-Ke Pan, and Ponnuthurai Nagaratnam Suganthan</i> | |
| Design of Non-uniformly Weighted and Spaced Circular Antenna Arrays with Reduced Side Lobe Level and First Null Beamwidth Using Seeker Optimization Algorithm | 35 |
| <i>Gopi Ram, Durbadal Mandal, Sakti Prasad Ghoshal, and Rajib Kar</i> | |
| Covariance Matrix Adaptation Evolutionary Strategy for the Solution of Transformer Design Optimization Problem | 47 |
| <i>Selvaraj Tamilselvi and Subramanian Baskar</i> | |
| Load Information Based Priority Dependant Heuristic for Manpower Scheduling Problem in Remanufacturing | 59 |
| <i>Shantanab Debchoudhury, Debrota Basu, Kai-Zhou Gao, and Ponnuthurai Nagaratnam Suganthan</i> | |
| A Tree Based Chemical Reaction Optimization Algorithm for QoS Multicast Routing | 68 |
| <i>Satya Prakash Sahoo, Sumaiya Ahmed, Manoj Kumar Patel, and Manas Ranjan Kabat</i> | |
| A New Improved Knowledge Based Cultural Algorithm for Reactive Power Planning | 78 |
| <i>Bidishna Bhattacharya, Kamal K. Mandal, and Niladri Chakraborty</i> | |
| BFO-RLDA: A New Classification Scheme for Face Images Using Probabilistic Reasoning Model | 88 |
| <i>Lingraj Dora, Sanjay Agrawal, and Rutuparna Panda</i> | |

| | |
|--|-----|
| Optimal Stable IIR Low Pass Filter Design Using Modified Firefly Algorithm | 98 |
| <i>Suman K. Saha, Rajib Kar, Durbadal Mandal, and Sakti Prasad Ghoshal</i> | |
| Firefly Algorithm with Various Randomization Parameters: An Analysis | 110 |
| <i>Nadaradjane Sri Madhava Raja, K. Suresh Manic, and V. Rajinikanth</i> | |
| Reducing Power Losses in Power System by Using Self Adaptive Firefly Algorithm | 122 |
| <i>B. Suresh Babu and A. Shunmugalatha</i> | |
| A Soft-Computing Based Approach to Economic and Environmental Analysis of an Autonomous Power Delivery System Utilizing Hybrid Solar – Diesel – Electrochemical Generation | 133 |
| <i>Trina Som and Niladri Chakraborty</i> | |
| Parameter Adaptation in Differential Evolution Based on Diversity Control | 146 |
| <i>S. Miruna Joe Amali and Subramanian Baskar</i> | |
| Data Clustering with Differential Evolution Incorporating Macromutations | 158 |
| <i>Goran Martinović and Dražen Bajer</i> | |
| Improved Adaptive Differential Evolution Algorithm with External Archive | 170 |
| <i>Rammohan Mallipeddi and Ponnuthurai Nagaratnam Suganthan</i> | |
| Fuzzy Clustering of Image Pixels with a Fitness-Based Adaptive Differential Evolution | 179 |
| <i>Soham Sarkar, Gyana Ranjan Patra, Swagatam Das, and Sheli Sinha Chaudhuri</i> | |
| Performance Study of a New Modified Differential Evolution Technique Applied for Optimal Placement and Sizing of Distributed Generation . . . | 189 |
| <i>S. Kumar, D. Pal, Kamal K. Mandal, and Niladri Chakraborty</i> | |
| An Approach to Solve Multi-criteria Supplier Selection While Considering Environmental Aspects Using Differential Evolution | 199 |
| <i>Sunil Kumar Jauhar, Millie Pant, and Aakash Deep</i> | |
| Comparison between Differential Evolution Algorithm and Particle Swarm Optimization for Market Clearing with Voltage Dependent Load Models | 209 |
| <i>Deep Kiran, Bijaya Ketan Panigrahi, and A.R. Abhyankar</i> | |

| | |
|---|-----|
| Multipopulation-Based Differential Evolution with Speciation-Based Response to Dynamic Environments | 222 |
| <i>Souwik Kundu, Debabrota Basu, Sheli Sinha Chaudhuri</i> | |
| A Modified Differential Evolution for Symbol Detection in MIMO-OFDM System | 236 |
| <i>Aritra Sen, Subhrajit Roy, and Swagatam Das</i> | |
| Lévy Flight Based Local Search in Differential Evolution | 248 |
| <i>Harish Sharma, Shimpi Singh Jadon, Jagdish Chand Bansal, and K.V. Arya</i> | |
| An Adaptive Differential Evolution Based Fuzzy Approach for Edge Detection in Color and Grayscale Images | 260 |
| <i>Satrajit Mukherjee, Bodhisattwa Prasad Majumder, Aritran Piplai, and Swagatam Das</i> | |
| A Differential Evolution Approach to Multi-level Image Thresholding Using Type II Fuzzy Sets | 274 |
| <i>Ritambhar Burman, Sujoy Paul, and Swagatam Das</i> | |
| Differential Evolution with Controlled Annihilation and Regeneration of Individuals and a Novel Mutation Scheme | 286 |
| <i>Sudipto Mukherjee, Sarthak Chatterjee, Debdipta Goswami, and Swagatam Das</i> | |
| Differential Evolution and Offspring Repair Method Based Dynamic Constrained Optimization | 298 |
| <i>Kunal Pal, Chiranjib Saha, and Swagatam Das</i> | |
| Adaptive Differential Evolution with Difference Mean Based Perturbation for Practical Engineering Optimization Problems | 310 |
| <i>Rupam Kundu, Rohan Mukherjee, and Swagatam Das</i> | |
| Transmission Line Management Using Multi-objective Evolutionary Algorithm | 321 |
| <i>K. Pandiarajan and C.K. Babulal</i> | |
| Normalized Normal Constraint Algorithm Based Multi-Objective Optimal Tuning of Decentralised PI Controller of Nonlinear Multivariable Process – Coal Gasifier | 333 |
| <i>Rangasamy Kotteeswaran and Lingappan Sivakumar</i> | |
| Simulated Annealing Based Real Power Loss Minimization Aspect for a Large Power Network | 345 |
| <i>Syamasree Biswas (Raha), Kamal Krishna Manadal, and Niladri Chakraborty</i> | |

| | |
|---|-----|
| Hybrid Artificial Bee Colony Algorithm and Simulated Annealing Algorithm for Combined Economic and Emission Dispatch Including Valve Point Effect | 354 |
| <i>Sundaram Arunachalam, R. Saranya, and N. Sangeetha</i> | |
| Spectrum Allocation in Cognitive Radio Networks Using Firefly Algorithm | 366 |
| <i>Kiran Kumar Anumandla, Shravan Kudikala, Bharadwaj Akella Venkata, and Samrat L. Sabat</i> | |
| Bi-objective Optimization in Identical Parallel Machine Scheduling Problem | 377 |
| <i>Sankaranarayanan Bathrinath, S. Saravana Sankar, S.G. Ponnambalam, and B.K.V. Kannan</i> | |
| Teaching-Learning-Based Optimization Algorithm in Dynamic Environments | 389 |
| <i>Feng Zou, Lei Wang, Xinhong Hei, Qiaoyong Jiang, and Dongdong Yang</i> | |
| A Novel Ant Colony Optimization Algorithm for the Vehicle Routing Problem | 401 |
| <i>Srinjoy Ganguly and Swagatam Das</i> | |
| Implementation of Fractional Order PID Controller for Three Interacting Tank Process Optimally Tuned Using Bee Colony Optimization | 413 |
| <i>U. Sabura Banu</i> | |
| Artificial Bee Colony-Based Approach for Optimal Capacitor Placement in Distribution Networks | 424 |
| <i>Attia El-Fergany, Almoataz Y. Abdelaziz, and Bijaya Ketan Panigrahi</i> | |
| Grammatical Bee Colony | 436 |
| <i>Tapas Si, Arunava De, and Anup Kumar Bhattacharjee</i> | |
| Artificial Bee Colony Algorithm for Probabilistic Target Q-coverage in Wireless Sensor Networks | 446 |
| <i>S. Mini, Siba K. Udgata, and Samrat L. Sabat</i> | |
| Chaos Synchronization in Commensurate Fractional Order Lü System via Optimal $PI^\lambda D^\mu$ Controller with Artificial Bee Colony Algorithm | 457 |
| <i>Anguluri Rajasekhar, Shantanu Das, and Swagatam Das</i> | |
| Cooperative Micro Artificial Bee Colony Algorithm for Large Scale Global Optimization Problems | 469 |
| <i>Anguluri Rajasekhar and Swagatam Das</i> | |

| | |
|---|-----|
| Improvement in Genetic Algorithm with Genetic Operator Combination (GOC) and Immigrant Strategies for Multicast Routing in Ad Hoc Networks | 481 |
| <i>P. Karthikeyan and Subramanian Baskar</i> | |
| Ensemble of Dying Strategies Based Multi-objective Genetic Algorithm..... | 492 |
| <i>Rahila Patel, M.M. Raghuvanshi, and L.G. Malik</i> | |
| Effect of Photovoltaic and Wind Power Variations in Distribution System Reconfiguration for Loss Reduction Using Ant Colony Algorithm..... | 504 |
| <i>H.A. Abdelsalam, Almoataz Y. Abdelaziz, R.A. Osama, and Bijaya Ketan Panigrahi</i> | |
| Inter-species Cuckoo Search via Different Levy Flights | 515 |
| <i>Swagatam Das, Preetam Dasgupta, and Bijaya Ketan Panigrahi</i> | |
| Cuckoo Search Algorithm for the Mobile Robot Navigation | 527 |
| <i>Prases Kumar Mohanty and Dayal R. Parhi</i> | |
| Automatic Generation Control of Multi-area Power System Using Gravitational Search Algorithm | 537 |
| <i>Rabindra Kumar Sahu, Umesh Kumar Rout, and Sidhartha Panda</i> | |
| Design and Simulation of FIR High Pass Filter Using Gravitational Search Algorithm | 547 |
| <i>R. Islam, Rajib Kar, Durbadal Mandal, and Sakti Prasad Ghoshal</i> | |
| Solution of Optimal Reactive Power Dispatch by an Opposition-Based Gravitational Search Algorithm | 558 |
| <i>Binod Shaw, V. Mukherjee, and Sakti Prasad Ghoshal</i> | |
| A Novel Swarm Intelligence Based Gravitational Search Algorithm for Combined Economic and Emission Dispatch Problems | 568 |
| <i>Hari Mohan Dubey, Manjaree Pandit, Bijaya Ketan Panigrahi, and Mugdha Udgir</i> | |
| Particle Swarm Optimization Based Optimal Reliability Design of Composite Electric Power System Using Non-sequential Monte Carlo Sampling and Generalized Regression Neural Network | 580 |
| <i>R. Ashok Bakkiyaraj and Narayanan Kumarappan</i> | |
| A Bacteria Foraging-Particle Swarm Optimization Algorithm for QoS Multicast Routing..... | 590 |
| <i>Rohini Pradhan, Manas Ranjan Kabat, and Satya Prakash Sahoo</i> | |
| Performance Evaluation of Particle Swarm Optimization Algorithm for Optimal Design of Belt Pulley System | 601 |
| <i>Pandurengan Sabarinath, M.R. Thansekhar, and R. Saravanan</i> | |

| | |
|---|-----|
| Optimal Sizing for Stand-Alone Hybrid PV-WIND Power Supply System Using PSO | 617 |
| <i>D. Suchitra, R. Jegatheesan, M. Umamaheswara Reddy, and T.J. Deepika</i> | |
| A Peer-to-Peer Dynamic Single Objective Particle Swarm Optimizer | 630 |
| <i>Hrishikesh Dewan, Raksha B. Nayak, and V. Susheela Devi</i> | |
| Aligned PSO for Optimization of Image Processing Methods Applied to the Face Recognition Problem | 642 |
| <i>Juan Luis Fernández-Martínez, Ana Cernea, Esperanza García-Gonzalo, Julian Velasco, and Bijaya Ketan Panigrahi</i> | |
| Optimal Operation Management of Transmission System with Fuel Cell Power Plant Using PSO | 652 |
| <i>S. Vidyasagar, K. Vijayakumar, and D. Sattianadan</i> | |
| PID Tuning and Control for 2-DOF Helicopter Using Particle Swarm Optimization | 662 |
| <i>A.P.S. Ramalakshmi, P.S. Manoharan, and P. Deepamangai</i> | |
| Optimal Location and Parameter Selection of Thyristor Controlled Series Capacitor Using Particle Swarm Optimization | 673 |
| <i>S. Devi and M. Geethanjali</i> | |
| A New Particle Swarm Optimization with Population Restructuring Based Multiple Population Strategy | 688 |
| <i>Qingjian Ni, Cen Cao, and Huimin Du</i> | |
| Small Signal Stability Constrained Optimal Power Flow Using Swarm Based Algorithm | 699 |
| <i>Mani Devesh Raj and Periyasami Somasundaram</i> | |
| Online Voltage Stability Assessment of Power System by Comparing Voltage Stability Indices and Extreme Learning Machine | 710 |
| <i>M.V. Suganyadevi and C.K. Babulal</i> | |
| A Peer-to-Peer Particle Swarm Optimizer for Multi-objective Functions | 725 |
| <i>Hrishikesh Dewan, Raksha B. Nayak, and V. Susheela Devi</i> | |
| A Novel Improved Discrete ABC Algorithm for Manpower Scheduling Problem in Remanufacturing | 738 |
| <i>Debabrota Basu, Shantanab Debchoudhury, Kai-Zhou Gao, and Ponnuthurai Nagaratnam Suganthan</i> | |

| | |
|--|-----|
| Optimal Partial-Retuning of Decentralised PI Controller of Coal Gasifier Using Bat Algorithm | 750 |
| <i>Rangasamy Kotteeswaran and Lingappan Sivakumar</i> | |
| Optimal Velocity Requirements for Earth to Venus Mission Using Taboo Evolutionary Programming | 762 |
| <i>M. Mutyalarao, Amaranathan Sabarinath, and M. Xavier James Raj</i> | |
| Author Index | 773 |

Table of Contents – Part II

| | |
|--|-----|
| Critical Issues in Model-Based Surrogate Functions in Estimation of Distribution Algorithms | 1 |
| <i>Roberto Santana, Alexander Mendiburu, and Jose A. Lozano</i> | |
| Sandholm Algorithm with K-means Clustering Approach for Multi-robot Task Allocation | 14 |
| <i>Murugappan Elango, Ganesan Kanagaraj, and S.G. Ponnambalam</i> | |
| Genetic Programming for Modeling Vibratory Finishing Process: Role of Experimental Designs and Fitness Functions | 23 |
| <i>Akhil Garg and Kang Tai</i> | |
| Non-fragile Robust PI Controller Design Using Co-variance Matrix Adaptation Evolutionary Strategy | 32 |
| <i>K. Mohaideen Abdul Kadhar and S. Baskar</i> | |
| Meta Heuristic Approaches for Circular Open Dimension Problem | 44 |
| <i>N. Madhu Sudana Rao, M. Aruna, and S. Bhuvaneshwari</i> | |
| Protein Function Prediction Using Adaptive Swarm Based Algorithm | 55 |
| <i>Archana Chowdhury, Amit Konar, Pratyusha Rakshit, and Ramadoss Janarthanan</i> | |
| Reduction of Bullwhip Effect in Supply Chain through Improved Forecasting Method: An Integrated DWT and SVM Approach | 69 |
| <i>Sanjita Jaipuria and S.S. Mahapatra</i> | |
| An Ant Colony Optimization Algorithm for the Min-Degree Constrained Minimum Spanning Tree Problem | 85 |
| <i>V. Venkata Ramana Murthy and Alok Singh</i> | |
| Multiobjective Differential Evolution Algorithm Using Binary Encoded Data in Selecting Views for Materializing in Data Warehouse | 95 |
| <i>Rajib Goswami, Dhruva Kumar Bhattacharyya, and Malayananda Dutta</i> | |
| Robust Protective Relay Setting and Coordination Using Modified Differential Evolution Considering Different Network Topologies | 107 |
| <i>Joymala Moirangthem, Bijaya Ketan Panigrahi, Krishnanand K.R., and Sanjib Kumar Panda</i> | |

| | |
|---|-----|
| Real-Coded Genetic Algorithm and Fuzzy Logic Approach for Real-Time Load-Tracking Performance of an Autonomous Power System | 119 |
| <i>Abhik Banerjee, V. Mukherjee, and S.P. Ghoshal</i> | |
| Short Term Load Forecasting (STLF) Using Generalized Neural Network (GNN) Trained with Adaptive GA | 132 |
| <i>D.K. Chaturvedi and Sinha Anand Premdayal</i> | |
| Gene Selection Using Multi-objective Genetic Algorithm Integrating Cellular Automata and Rough Set Theory | 144 |
| <i>Soumen Kumar Pati, Asit Kumar Das, and Arka Ghosh</i> | |
| Fusion at Features Level in CBIR System Using Genetic Algorithm..... | 156 |
| <i>Chandrashekhar G. Patil, Mahesh T. Kolte, and Devendra S. Chaudhari</i> | |
| New Bio-inspired Meta-Heuristics - Green Herons Optimization Algorithm - for Optimization of Travelling Salesman Problem and Road Network | 168 |
| <i>Chiranjib Sur and Anupam Shukla</i> | |
| A Comparative Analysis of Results of Data Clustering with Variants of Particle Swarm Optimization | 180 |
| <i>Anima Naik, Suresh Chandra Satapathy, and K. Parvathi</i> | |
| Hybrid Particle Swarm Optimization Technique for Protein Structure Prediction Using 2D Off-Lattice Model | 193 |
| <i>Nanda Dulal Jana and Jaya Sil</i> | |
| Software Effort Estimation Using Functional Link Neural Networks Optimized by Improved Particle Swarm Optimization | 205 |
| <i>Tirimula Rao Benala, Rajib Mall, and Satchidananda Dehuri</i> | |
| Improved Feature Selection Based on Particle Swarm Optimization for Liver Disease Diagnosis | 214 |
| <i>Gunasundari Selvaraj and Janakiraman S.</i> | |
| Groundwater System Modeling for Pollution Source Identification Using Artificial Neural Network | 226 |
| <i>Raj Mohan Singh and Divya Srivastava</i> | |
| Stochastic Analysis for Forecasting the MW Load of Plug-In Electric Vehicles | 237 |
| <i>C.S. Indulkar and K. Ramalingam</i> | |

| | |
|--|-----|
| Biometric Based Personal Authentication Using Eye Movement Tracking | 248 |
| <i>Atul Dhingra, Amioy Kumar, Madasu Hanmandlu, and Bijaya Ketan Panigrahi</i> | |
| Accelerated Simulation of Membrane Computing to Solve the N-queens Problem on Multi-core | 257 |
| <i>Ali Maroosi and Ravie Chandren Muniyandi</i> | |
| A Genetic Algorithm Optimized Artificial Neural Network for the Segmentation of MR Images in Frontotemporal Dementia | 268 |
| <i>R. Sheela Kumari, Tinu Varghese, C. Kesavadas, N. Albert Singh, and P.S. Mathuranath</i> | |
| Intelligent Computation and Kinematics of 4-DOF SCARA Manipulator Using ANN and ANFIS | 277 |
| <i>Panchanand Jha and Bibhuti Bhusan Biswal</i> | |
| Hybrid Neuro-Fuzzy Network Identification for Autonomous Underwater Vehicles | 287 |
| <i>Osama Hassanein, G. Sreenatha, and Tapabrata Ray</i> | |
| Prediction of Protein Structural Class by Functional Link Artificial Neural Network Using Hybrid Feature Extraction Method | 298 |
| <i>Bishnupriya Panda, Ambika Prasad Mishra, Babita Majhi, and Minakhi Rout</i> | |
| ANN Modeling of a Steelmaking Process | 308 |
| <i>Dipak Laha</i> | |
| Crop Yield Forecasting Using Neural Networks | 319 |
| <i>Mukesh Meena and Pramod Kumar Singh</i> | |
| Non-linear Dynamic System Identification Using FLLWNN with Novel Learning Method | 332 |
| <i>Mihir Narayan Mohanty, Badrinarayan Sahu, Prasanta Kumar Nayak, and Laxmi Prasad Mishra</i> | |
| Analysis of Transient Stability Based STATCOM for Neural Network Controller in Cascaded Multilevel Inverter | 342 |
| <i>P.K. Dhal and C. Christober Asir Ranjan</i> | |
| Emotion Recognition System by Gesture Analysis Using Fuzzy Sets | 354 |
| <i>Reshma Kar, Aruna Chakraborty, Amit Konar, and Ramadoss Janarthanan</i> | |
| Automatic Eye Detection in Face Images for Unconstrained Biometrics Using Genetic Programming | 364 |
| <i>Chandrashekhhar Padole and Joanne Athaide</i> | |

| | |
|--|-----|
| Neural Network Based Gesture Recognition for Elderly Health Care Using Kinect Sensor | 376 |
| <i>Sriparna Saha, Monalisa Pal, Amit Konar, and Ramadoss Janarthanan</i> | |
| City Block Distance for Identification of Co-expressed MicroRNAs | 387 |
| <i>Sushmita Paul and Pradipta Maji</i> | |
| Extreme Learning Machine Approach for On-Line Voltage Stability Assessment | 397 |
| <i>P. Duraipandy and D. Devaraj</i> | |
| Quadratic Fuzzy Bilevel Chance Constrained Programming with Parameters Following Weibull Distribution | 406 |
| <i>Animesh Biswas and Arnab Kumar De</i> | |
| Message Passing Methods for Estimation of Distribution Algorithms Based on Markov Networks | 419 |
| <i>Roberto Santana, Alexander Mendiburu, and Jose A. Lozano</i> | |
| Application of Neural Networks to Automatic Load Frequency Control | 431 |
| <i>Soumyadeep Nag and Namitha Philip</i> | |
| RNN Based Solar Radiation Forecasting Using Adaptive Learning Rate | 442 |
| <i>Ajay Pratap Yadav, Avanish Kumar, and Laxmidhar Behera</i> | |
| Energy Efficient Aggregation in Wireless Sensor Networks for Multiple Base Stations | 453 |
| <i>Nagarjuna Reddy Busireddy and Siba K. Udgata</i> | |
| An Intelligent Method for Handoff Decision in Next Generation Wireless Network | 465 |
| <i>Laksha Pattnaik, Mihir Narayan Mohanty, and Bibhuprasad Mohanty</i> | |
| Path Planning for the Autonomous Underwater Vehicle | 476 |
| <i>Andrey Kirsanov, Sreenatha G. Anavatti, and Tapabrata Ray</i> | |
| A Game Theoretic Approach for Reliable Power Supply in Islanded DG Grids | 487 |
| <i>Rohan Mukherjee, Rupam Kundu, Sanjoy Das, Bijaya Ketan Panigrahi, and Swagatam Das</i> | |
| Classification of Day-Ahead Deregulated Electricity Market Prices Using DCT-CFNN | 499 |
| <i>S. Anbazhagan and Narayanan Kumarappan</i> | |
| Multi-Objective Approach for Protein Structure Prediction | 511 |
| <i>S. Sudha, S. Baskar, and S. Krishnaswamy</i> | |

| | |
|---|-----|
| Clustering Based Analysis of Spirometric Data Using Principal Component Analysis and Self Organizing Map | 523 |
| <i>Mythili Asaithambi, Sujatha C. Manoharan, and Srinivasan Subramanian</i> | |
| Feature Selection of Motor Imagery EEG Signals Using Firefly Temporal Difference Q-Learning and Support Vector Machine | 534 |
| <i>Saugat Bhattacharyya, Pratyusha Rakshit, Amit Konar, D.N. Tibarewala, and Ramadoss Janarthanan</i> | |
| Optimal Build-or-Buy Decision for Component Selection of Application Package Software | 546 |
| <i>P.C. Jha, Ramandeep Kaur, Shivani Bali, and Sushila Madan</i> | |
| Text and Data Mining to Detect Phishing Websites and Spam Emails | 559 |
| <i>Mayank Pandey and Vadlamani Ravi</i> | |
| Intelligent Fault Tracking by an Adaptive Fuzzy Predictor and a Fractional Controller of Electromechanical System – A Hybrid Approach | 574 |
| <i>Tribeni Prasad Banerjee and Swagatam Das</i> | |
| Differential Evolution and Bacterial Foraging Optimization Based Dynamic Economic Dispatch with Non-smooth Fuel Cost Functions | 583 |
| <i>Kanchapogu Vaisakh, Pillala Praveena, and Kothapalli Naga Sujatha</i> | |
| Permutation Flowshop Scheduling Problem Using Classical NEH, ILS-ESP Operator | 595 |
| <i>Vanita G. Tonge and Pravin Kulkarni</i> | |
| Analysis of Human Retinal Vasculature for Content Based Image Retrieval Applications | 606 |
| <i>Sivakamasundari J. and Natarajan V.</i> | |
| Activity Recognition Using Multiple Features, Subspaces and Classifiers | 617 |
| <i>M.M. Sardeshmukh, M.T. Kolte, and D.S. Chaudahri</i> | |
| Advanced Optimization by Progressive Mapping Search Method of PSO and Neural Network | 625 |
| <i>Dong Hwa Kim, Jin Ill Park, and X.Z. Gao</i> | |
| Optimal Placement of DG in Distribution System Using Genetic Algorithm | 639 |
| <i>D. Sattianadan, M. Sudhakaran, S.S. Dash, K. Vijayakumar, and P. Ravindran</i> | |

| | |
|--|-----|
| Intelligent Controllers in Path Tracking of a Manipulator with Bounded Disturbance Torque | 648 |
| <i>Neha Kapoor and Jyoti Ohri</i> | |
| Multiscale and Multilevel Wavelet Analysis of Mammogram Using Complex Neural Network | 658 |
| <i>E. Malar, A. Kandaswamy, and M. Gauthaam</i> | |
| Author Index | 669 |

A Populated Iterated Greedy Algorithm with Inver-Over Operator for Traveling Salesman Problem

M. Fatih Tasgetiren¹, Ozge Buyukdagli¹, Damla Kızılay¹, and Korhan Karabulut²

¹ Industrial Engineering Department, Yasar University, Izmir, Turkey

² Software Engineering Department, Yasar University, Izmir, Turkey

{fatih.tasgetiren, ozge.buyukdagli@yasar.edu.tr,

damla.kizilay@yasar.edu.tr, korhan.karabulut@yasar.edu.tr}

Abstract. In this study, we propose a populated iterated greedy algorithm with an Inver-Over operator to solve the traveling salesman problem. The iterated greedy (IG) algorithm is mainly based on the central procedures of destruction and construction. The basic idea behind it is to remove some solution components from a current solution and reconstruct them in the partial solution to obtain the complete solution again. In this paper, we apply this idea in a populated manner (IGP) to the traveling salesman problem (TSP). Since the destruction and construction procedure is computationally expensive, we also propose an iteration jumping to an Inver-Over operator during the search process. We applied the proposed algorithm to the well-known 14 TSP instances from TSPLIB. The computational results show that the proposed algorithm is very competitive to the recent best performing algorithms from the literature.

Keywords: traveling salesman problem, iterated greedy algorithm, inver-over operator, memetic algorithm, genetic algorithm, meta-heuristics.

1 Introduction

Symmetric traveling salesman problem (TSP) is well-known and widely studied combinatorial optimization problem which bases on the idea of finding the shortest tour between n cities that has to be visited once by the salesman. Euclidean TSP is a subset of TSP in which distances are on Euclidean plane. Mathematically speaking, the distance of any two vertices $u_i = (x_i, y_i)$ and $v_j = (x_j, y_j)$ is given by

$$d[u_i, v_j] = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

If a distance matrix $d[i, j]$ contains a distance from city i to city j , then the tour length is given by

$$f = \sum_{i=1}^{n-1} d[i, i+1] + d[n, 1] \quad (2)$$

TSP is stated as an NP-hard optimization problem by Arora [4]. In order to solve NP-hard combinatorial optimization problems, many heuristic or meta-heuristic algorithms have proposed: Simulated Annealing [8], [23], Tabu Search [14], Genetic

Algorithms [1], [19], [28], Variable Neighborhood Search [33], Iterated Local Search [3], Neural Networks [11], [26], Ant Colony Optimization [12], [52], Particle Swarm Optimization [10], [47], Harmony Search [16], Differential Evolution [36], Honey Bees Mating Optimization [30], Memetic Algorithm [11], [31].

TSP algorithms are classified as exact and approximate. Finding the optimal solution in a limited number of steps can be guaranteed by the exact ones such as cutting plane or facet finding algorithm [39] that has solved the large instances. Performance of the TSP can be evaluated by solution time and error value, and these exact algorithms are stated to have higher time complexity [21]. In recent years, for approximate algorithms, solutions have several percentage low error value, they are produced quickly and provide good solutions, but do not guarantee the optimal solution. In order to find approximate solutions some heuristic methods were arisen [12], [22], [32], [45-46], [48]. In the approximate algorithms, some of them have a small deviation from the optimal solution and if this can be accepted, it may be appropriate to use the approximate algorithm [2], [21]. The approximate algorithms for TSP can be examined in two different classes such as tour construction methods [5], [9], [42] and tour improvement methods [3], [17], [27]. According to tour construction method, new city is added at each step of a tour that is built but for the tour improvement method, an initial solution is generated and then this initial one is tried to be improved by applying the various exchanges. Several meta-heuristic algorithms are included in the tour improvement methods [8], [14], [19], [23], [31]. Very recently, a genetic algorithm with a Greedy Sub Tour Mutation (GSTM) is presented in [1] whereas a memetic algorithm with the improved Inver-Over operator is presented in [38]. In this paper, we present a populated iterated greedy (IGP) algorithm with Inver-Over operator (IGP_IO) to solve the traveling salesman problem (TSP). To the best of our knowledge, this is the first reported application of iterated greedy algorithm to the TSP in the literature.

The rest of the paper is organized as follows. In Section 2, iterated greedy algorithm with Inver-Over operator is given in detail. The computational results are given in Section 3. Finally, conclusions are given in Section 4.

2 Iterated Greedy Algorithm With Inver-Over Operator

As mentioned before, the basic idea behind the IG algorithm is to remove some solution components from a current solution and reconstruct them in the partial solution to obtain the complete solution again. In this paper, we apply this idea in a populated manner (IGP) to the traveling salesman problem (TSP). Since the destruction and construction procedure is computationally expensive even though we employ a speed-up procedure for insertion, we also propose an iteration jumping to an Inver-Over operator during the search process in order to take advantage of both algorithms. Following subsections describe the IG algorithm and the IO operator briefly, and then we describe the IG with the IO operator in detail.

2.1 Iterated Greedy Algorithm

The IG algorithm is presented in Ruiz and Stützle [43], which has successful applications in discrete/combinatorial optimization problems such as in [7], [13], [15], [18], [24-25], [29], [35], [37], [41], [44], [49], [51], [53]. The IG algorithm is fascinating in terms of its conceptual simplicity, which makes it easily tunable and extendible to any combinatorial optimization problem. In an IG algorithm, there are two central procedures consisting of the destruction and the construction phases. The algorithm starts from some initial solution and then iterates through a main loop where a partial candidate solution is first obtained by removing a number of solution components from a complete candidate solution. This is called the destruction phase. Next a complete solution is reconstructed with a constructive insertion heuristic by inserting each node in the partial candidate solution. Before continuing with the next loop, an acceptance criterion is then used to decide whether or not the re-constructed solution will replace the incumbent one. This is called construction phase. These simple steps are iterated until some predetermined termination criterion such as a maximum number of iterations or a computation time limit is met. An outline of the IG algorithm is given in Figure 1.

Procedure IG()

```

 $\pi = \text{GenerateInitialSolution}()$ 
 $\pi_{best} = \pi$ 
while (NotTermination) do
     $\pi_1 = \text{DestructConstruct}(\pi, d)$ 
    if ( $f(\pi_1) < f(\pi)$ ) then
         $\pi = \pi_1$ 
        if ( $f(\pi) < f(\pi_{best})$ ) then
             $\pi_{best} = \pi$ 
    endwhile
return  $\pi_{best}$ 
endprocedure

```

Fig. 1. Iterated Greedy Algorithm

Regarding the destruction and construction procedure denoted as $\text{DestructConstruct}(\pi, d)$, d nodes are randomly taken from the solution π without repetition in the destruction step. We use the tournament selection with size 2 to select each node in such a way that two edges are chosen and distances of these two edges are computed. Then the worst one with the first node is removed from the solution. In the construction phase, the second phase of the NEH insertion heuristic [34] is used to complete the solution. To do so, the first node is inserted into all possible $n - d + 1$ positions in the destructed solution π^D generating $n - d + 1$ partial solutions. Among these $n - d + 1$ partial solutions, the best partial solution with the minimum tour length is chosen and kept for the next iteration. Then the second node is considered and so on until a final solution is obtained.

These insertions are computationally very expensive especially for larger instances. However, insertion cost can be evaluated by a nearest-neighbor criterion as follows: If a node k is to be inserted in an edge (u, v) , the insertion cost can be achieved by $c = d_{uk} + d_{kv} - d_{uv}$. It significantly accelerates the insertions necessary for the construction phase of the IG algorithm.

2.2 Inver-Over Operator

Inver over (IO) operator is a state-of-the-art algorithm for the TSP and proposed by Tao and Michalewicz [49]. This operator is based on inversion moves and aims to further improve the solution quality while using the information obtained from the current population. This makes the algorithm very adaptive and efficient. Figure 2 provides a detailed description of the algorithm. The algorithm starts with a randomly selected city from the individual and then the second city, that the inversion will be applied until, is selected in two different ways. With probability p , algorithm randomly selects the second city for inversion from the current individual. If $(\text{rand}()) < p$, then another individual is randomly selected from the population and a part of the pattern (at least 2 cities) of that individual is used as a reference to obtain better solutions. In this work, we employ a tournament selection (TS) with size 2 to determine another individual from the population instead of choosing randomly.

```

Procedure InverOver( )
Initialize Population
while (NotTermination) do{
  for (i = 1 to NP)do{
     $S' = S_i$ 
    Select randomly a city  $c$  from  $S'$ 
    while (TRUE) do{
      if (rand() <  $p$ ) then{
        select city  $c'$  from remaining cities in  $S'$ 
      }else{
        select an individual from  $P$  with TS with size 2
        Assign  $c'$  the next city of city  $c$  in the newly selected individual
      }endif
      if (the next city or the previous city of city  $c$  in  $S'$  is  $c'$ ) then{
        exit from while loop
      }endif
      inverse the section from the next city of city  $c$  to city  $c'$  in  $S'$ 
       $c = c'$ 
    }endwhile
    if ( $f(S') < f(S_i)$ ) then{
       $S_i = S'$ 
    }endif
  }endfor
Endprocedure

```

Fig. 2. Inver-Over Operator

2.3 Iterated Greedy Algorithm with Inver-Over Operator

In the proposed populated iterated greedy algorithm with Inver-Over operator (IGP_IO), the main idea is to jump into another operator at each iteration of a generation with a jumping probability jP . If a uniform random number $rnd(0,1)$ is less than jP , then the IGP_IO algorithm jump into the IO operator, otherwise, IG is applied to the individual on hand. By doing so, the search process of the IG algorithm is enriched and computational time is reduced.

The IGP_IO algorithm starts with an initial population. The permutation representation is used. We employ the Nearest Neighbor (NN) heuristic proposed in [42]. Figure 3 shows the construction of the initial population.

```

procedure InitPop( )
  for  $i = 1$  to  $n$  do  $bSelected[i] = false$ 
  for  $i = 1$  to  $n$  do
     $\pi[1] = i$ 
     $bSelected[\pi[1]] = true$ 
    for  $k = 2$  to  $n$  do
       $min = INT\_MAX$ 
      for  $j = 1$  to  $n$  do
        if ( $bSelected[j] = true$ ) continue
        if ( $d[\pi[k - 1][j]] < min$ )
           $min = d[\pi[k - 1][j]]$ 
           $position = j$ 
        endif
      endfor
       $\pi[k] = position$ 
       $bSelected[\pi[k]] = true$ 
    endfor
    store  $\pi^i$ 
  endfor
  sort  $\pi^i, i = 1, 2, \dots, n$ 
  keep  $\pi^i, i = 1, 2, \dots, NP$ 
endprocedure

```

Fig. 3. Initial Population with NN Heuristic

In the *InitPop*() procedure, NN heuristic generates n solutions. We sort them according to their fitness values. Then we only take NP solutions for the initial population. This procedure provides a good initial population with sufficient diversity.

Then, for each individual in the population, we apply either the IG algorithm or the IO operator depending on the jumping probability jP . There are two critical parameters of the IGP_IO algorithm. These are the jumping probability jP and the destruction size d . After some experiments, the destruction size is taken as $< 0.25 \times n$. Regarding the jumping probability jP . We tried the following values as $jP = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. Then we decided to take the jumping

probability as $jP = 0.8$ as the details will be given in Section 3. In addition, the population size is fixed at $NP = 30$. A detailed computational procedure of the IGP_IO algorithm is given in Figure 4.

```

Procedure IGP( )
 $\omega = \text{InitPop}(\pi_1, \pi_2, \dots, \pi_{NP})$ 
 $F = (f(\pi_1), f(\pi_2), \dots, f(\pi_{NP}))$ 
 $\pi_{best} = \text{argmin}\{\omega\}$ 
do{
  for  $i = 1$  to  $NP$ {
    if  $\text{rnd}(0,1) < jP$  then{
       $\pi_1 = \text{Inver} - \text{Over}(\pi_i)$ 
    }else
       $\pi_1 = \text{DestructConstruct}(\pi_i, d)$ 
    }endif
    if  $f(\pi_1) < f(\pi_i)$  then{
       $\pi_i = \pi_1$ 
      if  $f(\pi_1) < f(\pi_{best})$  then{
         $\pi_{best} = \pi_1$ 
      }endif
    }endif
  }endif
}endif
return  $\pi_{best}$ 
EndProcedure

```

Fig. 4. Populated Iterated Greedy Algorithm with Inver-Over Operator

3 Computational Results

The proposed IGP_IO algorithm is coded in C++ and run on an Intel (R) Core (TM) i5 CPU with 2.67 GHz PC with 4 GB memory. We test the performance of the proposed algorithm 14 instances from the TSPLIB [40]. As mentioned before, the population size is fixed at $NP = 30$. The destruction size and the jumping probability are taken as $= 0.25 \times n$, $jP = 0.8$, respectively. Initial population is established by the Nearest Neighbor (NN) heuristic proposed in [42]. Each instance is run for $R=10$ independent replications with 2500 generations. In order to compare the results, an average relative percentage deviation is calculated for each $R=10$ runs by using the following equation:

$$\Delta_{\text{avg}} = \sum_{i=1}^R \left(\frac{(H_i - \text{Best}) \times 100}{\text{Best}} \right) / R \quad (3)$$

where H_i is the objective function value that is obtained in i^{th} run of each algorithm, Best is the optimal objective function value and R is the number of runs. We fixed the destruction size at $d = 0.25 \times n$ since larger than $d = 0.25 \times n$ was

computationally very expensive. However, we analyzed the impact of the jumping probability in the ranges of $jP = \{0.0, 0.1, \dots, 1.0\}$. The overall averages with ten replications of 14 benchmark problems are given in Table 1. Based on the results in Table 1, we choose the jumping probability as $jP = 0.8$.

Table 1. Overall averages of 14 benchmarks with jumping probabilities

| jP | Δ_{avg} | Δ_{min} | Δ_{max} | Δ_{std} | CPU_{avg} |
|------------|----------------|----------------|----------------|----------------|-------------|
| 0.0 | 1.52 | 1.19 | 1.90 | 0.24 | 11.85 |
| 0.1 | 1.04 | 0.71 | 1.42 | 0.24 | 12.53 |
| 0.2 | 0.86 | 0.54 | 1.16 | 0.21 | 11.65 |
| 0.3 | 0.74 | 0.47 | 1.02 | 0.18 | 10.10 |
| 0.4 | 0.68 | 0.39 | 1.00 | 0.20 | 9.90 |
| 0.5 | 0.67 | 0.45 | 0.97 | 0.18 | 7.44 |
| 0.6 | 0.65 | 0.36 | 0.88 | 0.17 | 5.68 |
| 0.7 | 0.63 | 0.41 | 0.85 | 0.15 | 4.26 |
| 0.8 | 0.66 | 0.35 | 0.95 | 0.20 | 3.61 |
| 0.9 | 0.78 | 0.53 | 1.03 | 0.16 | 2.59 |
| 1.0 | 6.81 | 5.83 | 7.58 | 0.54 | 1.75 |

In order to test the performance of the proposed IGP_IO algorithm, we first compare the IGP with and without IO operator. The computational results of pure IGP and IGP_IO algorithms are given in Table 2. As can be seen in Table 2, embedding the IO operator into the IGP algorithm improved the solution quality since all statistics are further improved. In other words, Δ_{avg} , Δ_{min} , Δ_{max} , Δ_{std} and CPU values are further improved from 1.52, 1.19, 1.99, 0.22 and 11.85 to 0.66, 0.35, 0.95, 0.20 and 3.61.

Table 2. Average relative percentage deviations of the algorithms

| | IGP | | | | | IGP_IO | | | | |
|------------|----------------|----------------|----------------|----------------|-------|----------------|----------------|----------------|----------------|-------------|
| | Δ_{avg} | Δ_{min} | Δ_{max} | Δ_{std} | CPU | Δ_{avg} | Δ_{min} | Δ_{max} | Δ_{std} | CPU |
| berlin52 | 0.00 | 0.00 | 0.00 | 0.00 | 0.23 | 0.00 | 0.00 | 0.00 | 0.00 | 0.16 |
| kroA100 | 0.01 | 0.00 | 0.05 | 0.02 | 1.58 | 0.00 | 0.00 | 0.00 | 0.00 | 0.47 |
| pr144 | 0.39 | 0.39 | 0.39 | 0.00 | 0.13 | 0.35 | 0.00 | 0.39 | 0.12 | 0.43 |
| ch150 | 2.55 | 2.33 | 2.91 | 0.16 | 2.43 | 0.30 | 0.25 | 0.32 | 0.04 | 0.87 |
| kroB150 | 1.06 | 0.76 | 1.38 | 0.20 | 5.92 | 0.84 | 0.58 | 1.12 | 0.15 | 2.09 |
| pr152 | 0.44 | 0.18 | 0.77 | 0.27 | 5.28 | 0.51 | 0.00 | 0.63 | 0.18 | 1.03 |
| rat195 | 3.24 | 2.80 | 3.53 | 0.27 | 4.54 | 1.48 | 1.08 | 1.98 | 0.32 | 2.83 |
| d198 | 1.48 | 0.74 | 2.02 | 0.46 | 10.60 | 0.65 | 0.43 | 1.06 | 0.22 | 3.50 |
| kroA200 | 2.33 | 1.94 | 3.33 | 0.45 | 8.98 | 0.52 | 0.34 | 0.75 | 0.14 | 3.77 |
| ts225 | 2.46 | 2.46 | 2.46 | 0.00 | 2.76 | 0.20 | 0.00 | 0.63 | 0.28 | 3.14 |
| pr226 | 0.13 | 0.00 | 0.53 | 0.21 | 12.75 | 0.33 | 0.00 | 0.72 | 0.26 | 3.53 |
| pr299 | 2.82 | 2.04 | 3.41 | 0.50 | 22.84 | 1.04 | 0.28 | 1.82 | 0.56 | 6.21 |
| lin318 | 2.16 | 1.45 | 2.65 | 0.39 | 29.18 | 1.60 | 0.94 | 2.02 | 0.29 | 7.12 |
| pcb442 | 2.24 | 1.55 | 3.15 | 0.45 | 58.70 | 1.42 | 0.98 | 1.81 | 0.26 | 15.33 |
| <i>Avg</i> | 1.52 | 1.19 | 1.90 | 0.24 | 11.85 | 0.66 | 0.35 | 0.95 | 0.20 | 3.61 |

We provide the interval plot of all algorithms in Figure 5. From Figure 5, it can be seen that IGP_IO and IGP algorithms were statistically not equivalent because their confidence intervals does not coincide.

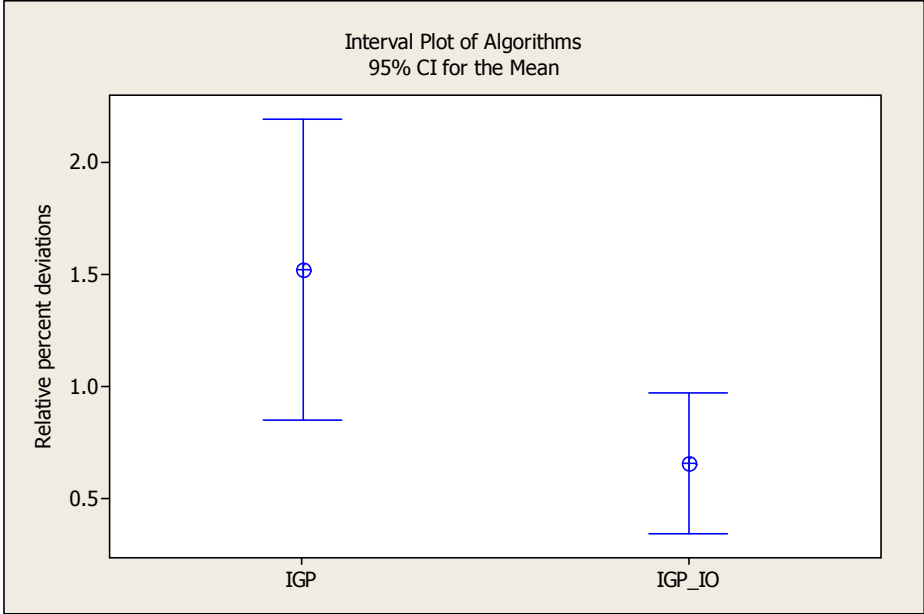
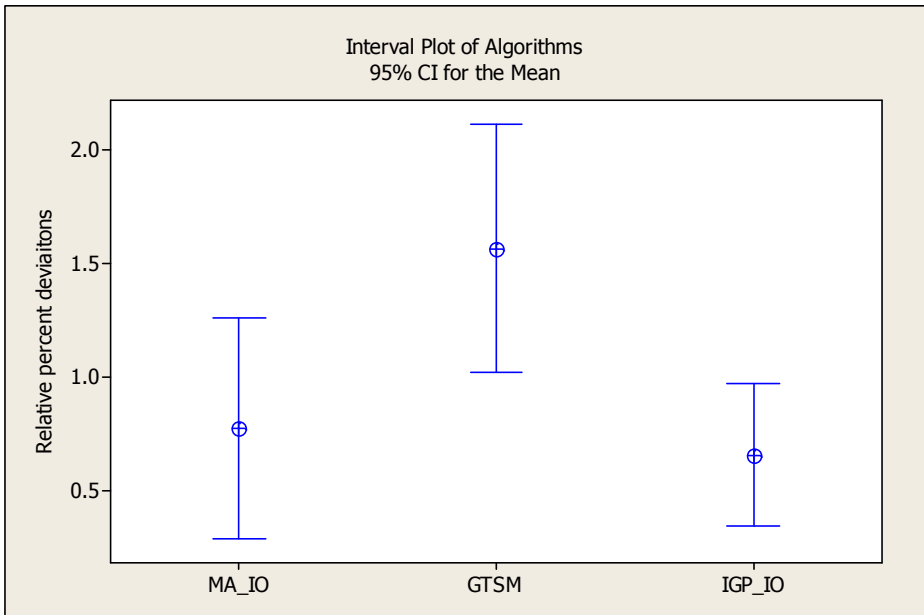


Fig. 5. Interval plots of algorithms compared.

To further test the performance of the IGP_IO algorithm, we compare it to the very recent two algorithms from the literature. These are the memetic algorithm based on the improved Inver-over operator denoted as MA_IO [38] and a genetic algorithm with Greedy Sub Tour Mutation (GSTM) [1]. Note that excellent results are provided in [38] with Lin-Kernighan (LK) local search. However, we only compare to MA_IO without LK local search in [38] since we do not use any local search algorithm. The computational results are given in Table 3. As can be seen in Table 3, IGP_IO operator was superior to GSTM algorithm since Δ_{avg} and Δ_{min} values were decreased from 1.57 and 0.66 to 0.66 and 0.35. IGP_IO algorithm was also computationally less expensive than the GSTM algorithm. When compared to the MA_IO algorithm, IGP_IO algorithm performed very similar to the MA_IO algorithm since the overall Δ_{min} values were the same. However, the IGP_IO algorithm was slightly better than the MA_IO algorithm on the overall Δ_{avg} values. However, the differences were not statistically significant. In other words, the IGP_IO algorithm was as good as the MA_IO algorithm. In terms of CPU times, MA_IO algorithm was much faster than the proposed IGP_IO algorithm.

Table 3. Average relative percentage deviation of the algorithms

| | <i>MA_IO</i> | | | <i>GTSM</i> | | | <i>IGP_IO</i> | | |
|----------|----------------|----------------|------------|----------------|----------------|------------|----------------|----------------|-------------|
| | Δ_{avg} | Δ_{min} | <i>CPU</i> | Δ_{avg} | Δ_{min} | <i>CPU</i> | Δ_{avg} | Δ_{min} | <i>CPU</i> |
| berlin52 | 0.00 | 0.00 | 0.49 | 0.00 | 0.00 | 0.84 | 0.00 | 0.00 | 0.16 |
| kroA100 | 0.00 | 0.00 | 0.62 | 1.18 | 0.00 | 6.99 | 0.00 | 0.00 | 0.47 |
| pr144 | 0.14 | 0.06 | 0.69 | 1.08 | 0.00 | 13.60 | 0.35 | 0.00 | 0.43 |
| ch150 | 0.36 | 0.00 | 0.86 | 0.64 | 0.46 | 11.24 | 0.30 | 0.25 | 0.87 |
| kroB150 | 0.65 | 0.04 | 0.78 | 1.76 | 0.96 | 11.68 | 0.84 | 0.58 | 2.09 |
| pr152 | 0.13 | 0.00 | 0.71 | 1.62 | 0.77 | 7.94 | 0.51 | 0.00 | 1.03 |
| rat195 | 0.66 | 0.43 | 0.85 | 1.84 | 0.60 | 15.05 | 1.48 | 1.08 | 2.83 |
| d198 | 0.68 | 0.34 | 0.94 | 1.22 | 0.39 | 12.10 | 0.65 | 0.43 | 3.50 |
| kroA200 | 0.58 | 0.41 | 0.90 | 1.54 | 0.87 | 13.29 | 0.52 | 0.34 | 3.77 |
| ts225 | 0.49 | 0.00 | 0.98 | 0.50 | 0.25 | 11.56 | 0.20 | 0.00 | 3.14 |
| pr226 | 0.43 | 0.13 | 0.92 | 1.53 | 0.72 | 13.84 | 0.33 | 0.00 | 3.53 |
| pr299 | 2.34 | 0.67 | 1.19 | 2.92 | 1.23 | 17.42 | 1.04 | 0.28 | 6.21 |
| lin318 | 2.31 | 1.36 | 1.27 | 3.31 | 0.98 | 14.64 | 1.60 | 0.94 | 7.12 |
| pcb442 | 2.11 | 1.45 | 1.67 | 2.78 | 2.05 | 19.13 | 1.42 | 0.98 | 15.33 |
| Average | 0.78 | 0.35 | 0.92 | 1.57 | 0.66 | 12.09 | 0.66 | 0.35 | 3.61 |

**Fig. 6.** Interval plots of algorithms compared

We provide the interval plot of all algorithms in Figure 6. From Figure 6, it can be seen that IGP_IO algorithm was statistically better than GTSM algorithm because their confidence intervals does not coincide. However, it was equivalent to MA_IO algorithm since their confidence intervals coincide.

4 Conclusion

In this study, we propose an IGP_IO algorithm to solve the traveling salesman problem. To the best of our knowledge, this is the first reported application of iterated greedy algorithm to the TSP in the literature. Since the destruction and construction procedure is computationally expensive, we also propose an iteration jumping to an Inver-Over operator during the search process. The proposed algorithm was applied to the well-known 14 TSP instances from TSPLIB. The computational results show that the proposed algorithm was better than the GTSP algorithm and was very competitive to the MA_IO algorithm. For a future work, the IGP_IO algorithm will be applied to larger instances.

References

1. Albayrak, M., Allahverdi, N.: Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms. *Expert Syst. Appl.* 38, 1313–1320 (2011)
2. Albayrak, M.: Determination of route by means of Genetic Algorithms for printed circuit board driller machines. Master dissertation, p. 180, Selcuk University (2008)
3. Applegate, D., Cook, W., Rohe, A.: Chained Lin–Kernighan for large traveling salesman problems. *INFORMS J. Comput.* 15, 82–92 (2003)
4. Arora, S.: Polynomial-time approximation schemes for Euclidean TSP and other geometric problems. *J. ACM* 45, 753–782 (1998)
5. Bentley, J.L.: Fast algorithm for geometric traveling salesman problems. *ORSA J. Comput.* 4, 387–441 (1992)
6. Bentley, J.L.: Experiments on traveling salesman heuristics. In: *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 91–99 (1990)
7. Bouamama, S., Blum, C., Boukerram, A.: A population-based iterated greedy algorithm for the minimum weight vertex cover problem. *Appl. Soft. Comput. J.* (2012), doi:10.1016/j.asoc.2012.02.013
8. Chen, Y., Zhang, P.: Optimized annealing of traveling salesman problem from the nth-nearest-neighbor distribution. *Physica A* 371, 627–632 (2006)
9. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* 12, 568–581 (1964)
10. Clerc, M.: Discrete particle swarm optimization, illustrated by the traveling salesman problem. In: Onwubolu, G.C., Babu, B.V. (eds.) *New Optimization Techniques in Engineering*. STUDEFUZZ, vol. 141, pp. 219–239. Springer, Heidelberg (2004)
11. Créput, J.C., Koukam, A.: A Memetic neural network for the Euclidean traveling salesman problem. *Neurocomputing* 72, 1250–1264 (2009)
12. Dorigo, M., Gambardella, L.M.: Ant colonies for the traveling salesman problem. *BioSystems* 43, 73–81 (1997)
13. Fanjul-Peyro, L., Ruiz, R.: Iterated greedy local search methods for unrelated parallel machine scheduling. *European Journal of Operational Research* 207, 55–69 (2010)
14. Fiechter, C.-N.: A parallel Tabu search algorithm for large traveling salesman problems. *Discrete Appl. Math.* 51, 243–267 (1994)
15. Framinan, J.M., Leisten, R.: Total tardiness minimization in permutation flow shops: a simple approach based on a variable greedy algorithm. *International Journal of Production Research* 46(22), 6479–6498 (2008)

16. Geem, Z.W., Tseng, C.-L., Park, Y.-J.: Harmony search for generalized orienteering problem: best touring in China. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 741–750. Springer, Heidelberg (2005)
17. Gendreau, M., Hertz, A., Laporte, G.: New insertion and post optimization procedures for the traveling salesman problem. *Oper. Res.* 40, 1086–1094 (1992)
18. Gerardo Minella, G., Ruiz, R., Ciavotta, M.: Restarted Iterated Pareto Greedy algorithm for multi-objective flowshop scheduling problems. *Computers & Operations Research* 38, 1521–1533 (2011)
19. Tao, G., Michalewicz, Z.: Inver-over operator for the TSP. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 803–812. Springer, Heidelberg (1998)
20. Gutin, G., Punnen, A.P.: *The Traveling Salesman Problem and its Variations*. Kluwer Academic Publishers, Dordrecht (2002)
21. Helsgaun, K.: An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research* 126(1), 106–130 (2000)
22. Jayalakshmi, G.A., Sathiamoorthy, S., Rajaram, R.: A hybrid genetic algorithm – a new approach to solve traveling salesman problem. *International Journal of Computational Engineering Science* 2(2), 339–355 (2001)
23. Jeong, C.S., Kim, M.H.: Fast parallel simulated annealing for traveling salesman problem on SIMD machines with linear interconnections. *Parallel Comput.* 17, 221–228 (1991)
24. Kahraman, C., Engin, O., Kaya, I., Ozturk, R.E.: Multiprocessor task scheduling in multistage hybrid flow-shops: A parallel greedy algorithm approach. *Applied Soft Computing* 10, 1293–1300 (2010)
25. Kuo-Ching Ying, K.-C., Lin, S.-W., Huang, C.-Y.: Sequencing single-machine tardiness problems with sequence dependent setup times using an iterated greedy heuristic. *Expert Systems with Applications* 36, 7087–7092 (2009)
26. Leung, K.S., Jin, H.D., Xu, Z.B.: An expanding self-organizing neural network for the travelling salesman problem. *Neurocomputing* 62, 267–292 (2004)
27. Lin, S., Kernighan, B.W.: An effective heuristic algorithm for the traveling salesman problem. *Oper. Res.* 21, 498–516 (1973)
28. Louis, S.J., Li, G.: Case injected genetic algorithms for traveling salesman problems. *Inform. Sci.* 122, 201–225 (2000)
29. Lozano, M., Molina, D., Garcia-Martinez, C.: Iterated greedy for the maximum diversity problem. *European Journal of Operational Research* 214, 31–38 (2011)
30. Marinakis, Y., Marinaki, M., Dounias, G.: Honey bees mating optimization algorithm for the Euclidean traveling salesman problem. *Inform. Sci.* (2010), doi:10.1016/j.ins.2010.06.032
31. Merz, P., Freisleben, B.: Memetic algorithms for the traveling salesman problem. *Complex Systems* 13, 297–345 (2001)
32. Misevicius, A.: Using iterated tabu search for the traveling salesman problem. *Information Technology and Control* 3(32), 29–40 (2004)
33. Mladenović, N., Hansen, P.: Variable neighborhood search. *Comput. Oper. Res.* 24, 1097–1100 (1997)
34. Nawaz, M., Ensco Jr., E.E., Ham, I.A.: Heuristic algorithm for the m-machine, n-node flow shop sequencing problem. *OMEGA* 11(1), 91–95 (1983)
35. Qinma Kanga, Q., Heb, H., Song, H.: Task assignment in heterogeneous computing systems using an effective iterated greedy algorithm. *The Journal of Systems and Software* 84, 985–992 (2011)

36. Onwubolu, G.C.: Optimizing CNC drilling machine operations: traveling salesman problem-differential evolution approach. In: Onwubolu, G.C., Babu, B.V. (eds.) *New Optimization Techniques in Engineering*. STUDEFUZZ, vol. 141, pp. 537–565. Springer, Heidelberg (2004)
37. Pan, Q.-K., Fatih, T.M., Liang, Y.-C.: A discrete differential evolution algorithm for the permutation flowshop scheduling problem. *Computers & Industrial Engineering* 55, 795–816 (2008)
38. Pan, Q.-K., Wang, Y.-T., Li, J.-Q., Gao, K.-Z.: Memetic Algorithm based on improved Inver-Over operator and Lin-Kernighan local search for the euclidean traveling salesman problem. *Computers and Mathematics with Applications* 62, 2743–2754 (2011)
39. Padberg, M.W., Rinaldi, G.: A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Rev.* 33, 60–100 (1991)
40. Reinelt, G.: TSPLIB—a traveling salesman problem library. *ORSA J. Comput.* 3, 376–384 (1991)
41. Ribas, I., Ramon Companys, R., Tort-Martorell, X.: An iterated greedy algorithm for the flowshop scheduling problem with blocking. *Omega* 39, 293–301 (2011)
42. Rosenkrantz, D.J., Stearns, R.E., Lewis, P.M.: An analysis of several heuristics for the traveling salesman problem. *SIAM J. Comput.* 6, 563–581 (1977)
43. Ruiz, R., Stützle, T.: A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research* 177(3), 2033–2049 (2007)
44. Ruiz, R., Stützle, T.: An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research* 187, 1143–1159 (2008)
45. Schleuter, M.G.: Asparagos96 and the traveling salesman problem. In: *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, pp. 171–174. IEEE Press (1997)
46. Seo, D., Moon, B.: Voronoi quantized crossover for traveling salesman problem. In: *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 544–552 (2002)
47. Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C., Wang, Q.X.: Particle swarm optimization-based algorithms for TSP and generalized TSP. *Inform. Process. Lett.* 103, 169–176 (2007)
48. Stützle, T., Hoos, H.: The MAX-MIN ant system and local search for the traveling salesman problem. In: *Proceedings of the IEEE International Conference on Evolutionary Computation*, Indianapolis, Indiana, USA, pp. 308–313 (1997)
49. Tao, G., Michalewicz, Z.: Inver-over operator for the TSP. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) *PPSN 1998*. LNCS, vol. 1498, pp. 803–812. Springer, Heidelberg (1998)
50. Tasgetiren, M., Fatih, P.Q.-K., Suganthan, P.N., Chen Angela, H.-L.: A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops. *Information Sciences* 181, 3459–3475 (2011)
51. Tasgetiren, M., Fatih, P.Q.-K., Liang, Y.-C.: A discrete differential evolution algorithm for the single machine total weighted tardiness problem with sequence dependent setup times. *Computers & Operations Research* 36, 1900–1915 (2009)
52. Tsai, C.F., Tsai, C.W., Tseng, C.C.: A new hybrid heuristic approach for solving large traveling salesman problem. *Inform. Sci.* 166, 67–81 (2004)
53. Ying, K.-C., Cheng, H.-M.: Dynamic parallel machine scheduling with sequence-dependent setup times using an iterated greedy heuristic. *Expert Systems with Applications* 37, 2848–2852 (2010)

Meta-modeling and Optimization for Varying Dimensional Search Space

Kalyanmoy Deb¹, Soumil Srivastava², and Akshay Chawla²

¹ Michigan State University, East Lansing, MI 48864, USA
kdeb@egr.msu.edu

² Indian Institute of Technology Kanpur, PIN 208016, India
soumil.iitk@gmail.com, achawla@iitk.ac.in
<http://www.egr.msu.edu/~kdeb>

Abstract. High-fidelity computer simulations are used widely in several scientific and engineering domains to study, analyze and optimize process responses and reduce the time, cost and risk associated with conducting a physical experiment. However, many such simulations are computationally expensive and impractical for optimization. Meta-models have been successfully used to give quick approximation of the process responses in simulations and facilitate the analysis and optimization of designs.

Despite the abundance of literature in meta-modeling for continuous variables, there have been very few studies in the domain where the design spaces are discrete or mixed or with dependencies between discrete and real variables. These problems are widespread in engineering, science, economics and several other fields. Through this work, we wish to address the lack of a technique to handle such problems from front to end i.e. selecting design samples, meta-modeling and subsequent optimization.

This paper presents novel methods for choosing design samples, meta-modeling of design spaces having binary and real variables using padding in Kriging technique and single-objective constrained optimization of the meta-model using a new genetic algorithm VDGA. These scalable generic methodologies have the potential for solving optimization problems that are very expensive or impractical due to the extremely high computational cost and time associated with the simulations. We also present the results of these techniques on several test problems.

Keywords: Meta-modeling, optimization, mixed-variable design space, discrete design space, evolutionary algorithms.

1 Introduction

Since the advent of computers, several engineering problems have been studied using computer simulations instead of the real-life experiments. A major reason is the reduction in the time, costs and risk involved in the experiment, e.g. a car crash study. In cases where the simulations themselves are quite costly and time-taking, meta-models are used to predict the output of the simulations using a few design samples.

While much work in the field of meta-modeling and subsequent optimization has been done for real-variables, not many researchers have studied the problems with discrete-variable design space or mixed-variable design space (containing both real and discrete variables). Another issue is the scalability of the proposed methods considering that full-factorial design samples are not possible in many cases.

Several meta-modeling techniques have been proposed for real design variable space and have become popular over the last two decades -- multivariate adaptive regression splines (MARS) [8], radial basis functions (RBF) [11][7], Kriging [1], response surface methodology [2] and artificial neural networks (ANN) [3] to name a few. Each of them has their own features, pros and cons. The answer to the question that which of these techniques is better than the others is not an easy one. It depends upon the type of problem that one is dealing with. Simpson et al. [14] have compared Kriging with polynomial regression for multidisciplinary design optimization of aero spike nozzle. Yang et al. [15] compared enhanced MARS, Artificial Neural Networks, Stepwise Regression and Moving Least Square for safety functions in automotive crash analysis. A comprehensive comparison of MARS, RBF, polynomial regression and Kriging was done by Jin et al. [13] on multiple test problems with varying scale and non-linearity.

A notable point is that the above mentioned studies have been performed for real-variable design space and not much research has been done for problems in the discrete design space or mixed-variables design space, leave alone problems having varying dimensions. The study by Meckesheimer et al. [16] on a desk lamp performance model and Davis et al. [7] on process synthesis have addressed this domain of problems but both have used underlying methods which are not easily scalable. Meckesheimer et al. have used state-selecting meta-modeling approach which needs sufficient design points and can produce large errors near discontinuities in case of misclassification. Davis et al. have used a branch and bound framework which is known to be not-scalable. A recent study (Rykerk et al., 2012) considered a variable-dimensional problem and used various representation-recombination to point out useful combinations, but no meta-modeling was performed.

We wish to propose a scalable meta-modeling technique in this study to handle binary and real variable design spaces and optimize the meta-model using a genetic algorithm.

2 Portfolio Optimization Problem

An automobile company presented us with an optimization problem with the aim to optimize a vehicle *portfolio* which consisted of several nameplates and their trims in terms of the fuel economy, acceleration and selling price of the vehicles while satisfying the environmental regulation (PAFE) such that corporate objectives of contribution margin (CM) is maximized. Since we have the choice of not producing a certain nameplate if it is not economically viable, the variable set of this problem essentially consists of a set binary variables indicating the presence or absence of nameplates and dependent real variables indicating the fuel economy, acceleration and price of trims.

The expected shares of the nameplates in the market are evaluated by a simulation in AMIS (AMI Software) which is quite costly and time-taking and we would like to replace it via a meta-model.

The portfolio consisted of 3 nameplates with 2, 4 and 4 trims respectively. We can set the fuel economy, acceleration and price of the vehicles in the portfolio. The total number of variables for this problem was 16 (real) + 3 (binary). The objective of the problem was to maximize the contribution margin (CM) subject to satisfying a Portfolio Average Fuel Economy (PAFE) target which acted as the constraint. Also, there are some lower and upper bounds for the variables. For this problem, real variables are all dependent upon one or the other value of the binary variables which act as presence of nameplates in the portfolio.

2.1 Varying Dimension Problem

The term ‘varying dimension problem’ is used to define a mixed variable problem. In this type of problem, several of the real variables of the problem are not independent but directly depend upon the value of certain discrete variables (or decision or presence variables). We have considered these discrete variables as binary (or Boolean) variables. For example, in this problem, the binary variables indicate the presence or absence of a particular trim. Furthermore, there are some real variables corresponding to each of the binary variable (i.e. trim), viz. acceleration, fuel economy etc. associated with the trim.

If a certain trim is absent, all the real variables corresponding to that trim would not have any effect in the analysis of that particular nameplate and would hence be inconsequential. So, when a trim is absent, the dimension of the design space is reduced by the number of real variables that are attached to that trim. Therefore, different solution in the search space has different sizes (number of variables), thereby making the search space of varying dimension. It is not at all clear how gradients in such problems can be defined, as dimension of the search space is a variable itself. Thus, classical optimization methods may not be appropriate for handling such problems.

Dasgupta and McGregor [6] had proposed the idea of a structured Genetic Algorithm (sGA) for multi-layered structure of the GA chromosome which acts as long-term distributed memory within the population. This idea is a bit similar to the varying dimension problem mentioned above, as there are dependencies between variables in sGA too. But sGA was capable of dealing with binary strings only and did not have mixed variables as in our case. The sGA was also not explored to incorporate meta-models. Varying dimension problems are found in several fields and domains and to the best of our knowledge, such as in dealing with truss-structure optimization, composite laminate design problems etc. No well-formulated optimization technique is suggested to handle them using a meta-modeling technique.

3 Meta-modeling

We have suggested the use of *padding* in the Kriging procedure for meta-modeling the objectives and constraints that consist of real and binary variables. We define Kriging and padding procedure briefly before moving on to the optimization algorithm.

3.1 Kriging

The Kriging technique was first proposed and named after Daniel G. Krige for geostatistics [6]. Several people have forwarded his work. Sacks et al [10] applied Kriging to deterministic computer experiments called Design and Analysis of Computer Experiments (DACE), which is precisely the type of Kriging used by us in this work. Kriging has a general assumption that $y(x) = \hat{y} + \varepsilon$. Here ε denotes the residuals that are identically distributed normal random variables but they are a function of x itself. Thus, in a Kriging meta-model, the process response is modeled as $\hat{y}(x) = f(x) + Z(x)$, where $f(x)$ is a regression model and $Z(x)$ is stochastic process. We have used the DACE toolbox for Kriging which is available online for free [7]. Kriging is a quite well-known meta-modeling technique which can give quite accurate approximations with a few sampling points.

3.2 Padding

Kriging theory exists only for real and continuous variables. The discrete portions of our problem cannot be handled directly by Kriging procedure. For transferring the information conveyed by the discrete variables, we propose a padding technique which pre-processes data before building a meta-model on it. In very simple terms, the real variables, corresponding to the absent (i.e. zero) binary variables are padded to a value which is *foreign* to the design space of interest. Then, the discrete variables are dropped for subsequent meta-modeling. It is also important to normalize all the real variables before doing this so that we can define the same padding value for each dimension. A standard score is found for normalizing the data according to $z = (x - \mu)/\sigma$ where x is the variable to be normalized, μ is the mean of population and σ is the standard deviation of the population. After normalization process, the data ranges roughly between -1 to +1. So a value of -2 or smaller can serve as a good (foreign) padding value. We have done some parametric study on the padding value as well which is available in [12]. The findings were that values close to the lower negative limit of the normalized data (-2, as per above normalization) set work well.

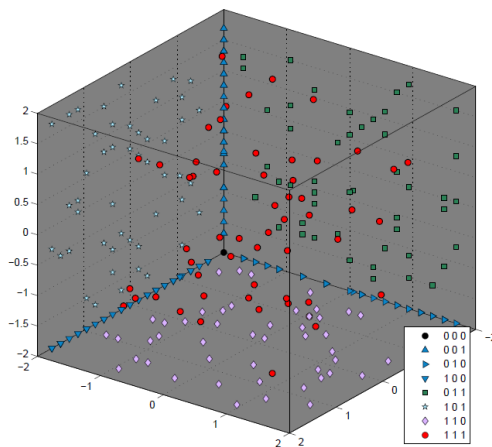


Fig. 1. The spatial segregation created by padding

The overall effect of this can be seen in an example displayed in figure 1. There are $2^3 = 8$ cases for 3 discrete variables. The [1 1 1] cases are mapped on the regular 3D space. The [1 1 0], [1 0 1] and [0 1 1] cases are mapped onto planes since the real variable corresponding to the 0 is padded to a fixed value. The cases [1 0 0], [0 1 0] and [0 0 1] are mapped onto lines and [0 0 0] case is just one point. Hence, all the cases are segregated effectively in the hyper-space and they do not interfere with each other. However, when we are not using the full-factorial design some of these planes or lines may be devoid of points. But we can still obtain some information from the nearby regions for a point of interest in such a region. Although it may not be very accurate, this allows us to build a meta-model that will give at least some information about such regions too. Also, this technique leads to a substantial reduction in the design space by removing all the discrete variables which reduces the size of the matrices inside Kriging and saves time, computational cost and error due to near-singularity of the same matrices.

4 Optimization of the Meta-model

Genetic Algorithms (GAs) are generic optimization algorithms and as already stated, they have been applied successfully in several fields. For the mixed-variable problems, GAs are ideal optimization technique. A major reason for this is that GAs do not use gradient information directly and gradients are not easily defined in discrete design spaces. Also, they can handle the discrete and real variables in their population members easily. We propose Varying Dimension Genetic Algorithm (VDGA) for handling the varying dimension problems. We have used tournament selection operator [9] with tournament of size or two for selection and the parameter-less approach for handling constraints [4]. For crossover and mutation, we have proposed the Varying Dimension Crossover and Mutation Operator (VDCM), which we shall discuss next. Also the SBX crossover [5] and adaptive polynomial mutation operator [4] have been used inside VDCM.

Before giving the details of VDCM, let us define a matrix which we call correspondence matrix or “corresmatrix”. This is used by the VDCM operator for crossover and mutation operations. Simply put, the “corresmatrix” is a matrix with rows for each binary variable and columns for each real variable. In each column, exactly one or none of elements is 1 and the rest are zeros. The binary variable row for which the element is 1 for a real variable column is the corresponding discrete variable for that particular real variable. For example if b_1 is the corresponding discrete variable for the real variable r_1 and b_2 of r_2 and so on, then the “corresmatrix” would look like one shown on right for a four-variable problem. A column with all zero elements means that the particular real variable is independent of all discrete variables. We now describe the proposed crossover and mutation operator.

| | r_1 | r_2 | r_3 | r_4 |
|-------|-------|-------|-------|-------|
| b_1 | 1 | 0 | 0 | 0 |
| b_2 | 0 | 1 | 0 | 0 |
| b_3 | 0 | 0 | 1 | 0 |
| b_4 | 0 | 0 | 0 | 1 |

4.1 VDCM (Varying Dimension Crossover and Mutation Operator) Operator

We now describe the VDCM operators used in this study.

1. Receive the “corresmatrix” and a mating pool of N members with n_b binary (discrete) variables and n_r real variables from the VDGA.
 2. Choose two parents p_1 and p_2 from the mating pool.
 3. Generate a random number between 0 and 1. If it is less than p_m (probability of mutation), set the “mutationflag” = 1 else set it as zero.
 4. Generate a random number between 0 and 1. If it is less than p_m (probability of crossover),
 - (a) Perform a *single-point crossover* on the discrete set of variables of the two parents and copy the resultant discrete sets in the two children c_1 and c_2 . Also copy the corresponding real variables for the non-exchanged part of the discrete set directly into the children. For the exchanged part of the discrete set, exchange the real variables also. Copy any independent real variables directly into the children.
 - (b) Now for each discrete variable b_i , search the “corresmatrix” to find its dependent real variables. Then, for each such real variable r_i ,
 - (i) if the values of the corresponding b_i is 1 in both c_1 and c_2 , perform a SBX crossover with the values copied in the children and replace them with the results. If “mutationflag” = 1, also do polynomial mutation individually on that real variable r_i in each child, else leave as it is.
 - (ii) if the values of the corresponding b_i is 1 in one child and 0 in the other, then if “mutationflag” = 1, perform a polynomial mutation on r_i in the child with corresponding $b_i = 1$, else leave as it is.
 - (iii) if the values of the corresponding b_i is 0 in both children, then leave the copied values as they are.
 - (c) For any independent real variables, do an SBX crossover. If “mutationflag” = 1, also do polynomial mutation individually on that real variable in each child else, leave as it is
- else,
- (d) Copy p_1 and p_2 into children c_1 and c_2 .
 - (e) If “mutationflag” = 1, then
 - (i) For each discrete variable b_i that is non-zero in c_1 , search the “corresmatrix” to find its dependent real variables and then, for each such real variable r_i , do polynomial mutation.
 - (ii) Repeat the above for c_2 too.
 - (iii) For any independent real variables in c_1 and c_2 , do polynomial mutation; else, continue.

5. Generate a random number between 0 and 1. If it is less than p_{mb} (probability of binary mutation), with a chance of 50% flip the binary variables of child c_1 .
6. Generate another random number between 0 and 1. If it is less than p_{mb} , with a chance of 50% flip the binary variables of child c_2 .
7. Go to Step 2 for choosing another set of two parents from the remaining members in the mating pool and continue until all parents are used.
8. Return the newly created N children to VDGA.

The probabilities of mutation p_m and binary mutation p_{mb} are set adaptively at each generation according to:

$$p_m = \frac{1}{n_r} + \frac{t}{t_{max}} \times \left(1 - \frac{1}{n_r}\right) \quad (1) \quad \text{and} \quad p_{mb} = 0.3 \times \left(\frac{1}{n_b} + \frac{t}{t_{max}} \times \left(1 - \frac{1}{n_b}\right)\right) \quad (2)$$

4.2 Varying Dimension Genetic Algorithm (VDGA)

The proposed VDGA is presented here:

1. Initialization

- (a) Set $t = 0$. Initialize the GA population P^t with N members that have n_b binary (discrete) variables and n_r real variables. The binary variables are made 0 or 1 with 50% probability and the real variables are initialized randomly between the defined lower and upper bounds.
- (b) Rank the population using the objective function and the penalty according to Deb's parameter-less approach. Note the best population member \mathbf{x}^t .
- (c) Increment t by one.

2. For every generation t , perform the following steps:

- (a) Perform one generation of the GA using objective function as the fitness measure to be minimized,
 - (i) Choose C elites (the best members) from the population and copy them to the offspring population Q^t
 - (ii) From the remaining members of P^t , select a mating pool M^t using binary tournament selection.
 - (iii) Use the VDCM operator on M^t to create a new offspring population R^t
 - (iv) Merge Q^t and R^t to form the population for the next generation P^{t+1}
- (b) Choose the best population member \mathbf{x}^t of P^{t+1} .
- (c) Increment counter t by one.

3. **Termination check:** If absolute difference between the objective function values of $\mathbf{x}^t < \varepsilon$, for τ generations, report the result of the last generation as the final result. Otherwise, increment counter t by one and go to Step 2.

5 Application on the Portfolio Optimization Problem

We present the results obtained for the optimization problem by the proposed algorithm in this section. Figures 2 and 3 compare the result obtained from the simulator and meta-model for the CM and PAFE, respectively.

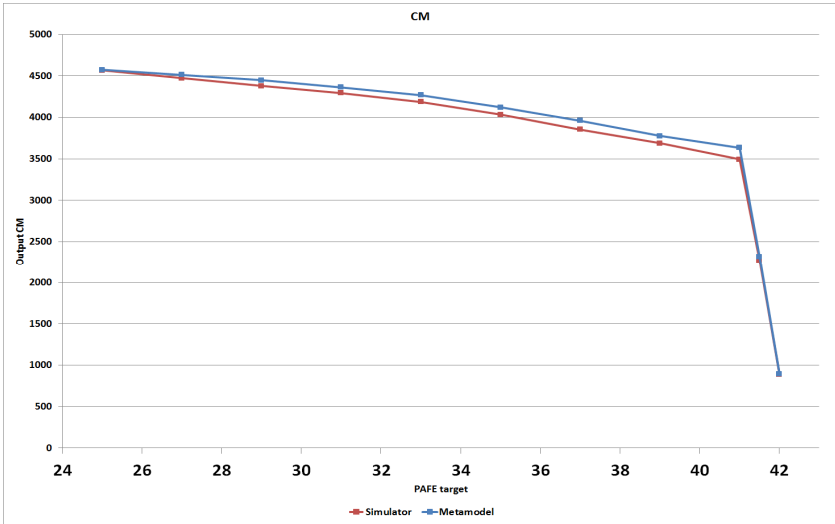


Fig. 2. Comparison of results obtained from the simulator and meta-model for CM

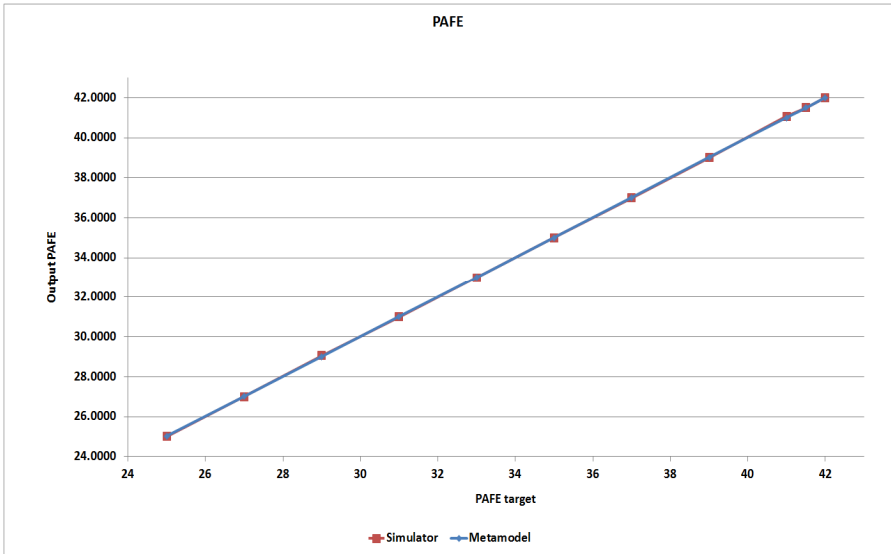


Fig. 3. Comparison of results obtained from the simulator and meta-model for PAFE

Table 1 presents the numerical values for the same, along with the difference between the two results.

Table 1. The numerical values for both PAFE and CM obtained from the meta-model and simulator

| Tar- get PAFE | CM (Meta- model) | CM (Simu- lator) | CM Error (%) | PAFE (Meta- model) | PAFE (Simu- lator) | PAFE Error (%) |
|------------------------------|---------------------------------|---------------------------------|-----------------------------|-----------------------------------|-----------------------------------|-------------------------------|
| 25 | 4574.077 | 4566.22 | 0.17 | 25.0214 | 25.0000 | 0.09 |
| 31 | 4362.448 | 4292.87 | 1.62 | 31.0116 | 31.0000 | 0.04 |
| 37 | 3962.028 | 3853.12 | 2.83 | 37.0151 | 37.0000 | 0.04 |
| 39 | 3774.945 | 3687.72 | 2.37 | 39.0167 | 39.0100 | 0.02 |
| 41 | 3632.855 | 3491.52 | 4.05 | 41.0012 | 41.0600 | -0.14 |
| 41.5 | 2316.653 | 2273.46 | 1.90 | 41.5001 | 41.5000 | 0.00 |
| 42 | 893.4042 | 886.69 | 0.76 | 42.0000 | 42.0100 | -0.02 |

The above results portray that the average error in case of CM is 1.84%, while in case of the PAFE it is -0.01%. As the error is small, we are able to mimic the simulator results with good accuracy. Therefore, the proposed methodology performs well for the given problem. The greatest advantage is achieved in terms of reduction in the time required to obtain the results. The simulator takes nearly 10 hours to obtain the results, while the meta-model is obtained in less than an hour.

6 Progressive Meta-modeling

With a large number of variables involved, meta-model building with a good accuracy on the entire search space may require a huge number of points and the process can then become computationally expensive and time consuming. Next, we used the idea of *progressive* modeling to overcome the aforementioned problems associated with the large-sized problem. We first select a small number of sample points, train a meta-model and then try to find regions with good objective function values. Then, we add more points from those regions to improve the accuracy of the meta-model. This also helps the optimization algorithm since if the actual simulation values of the points in that region are not as required, then the optimization algorithm will move away from those regions and search for better areas. The progressive meta-modeling technique is able to solve a large-sized problem. This particular problem consisted of 20 name-plates, and had 28 (real) + 20 (binary) variables. The results obtained are shown in Figure 4.

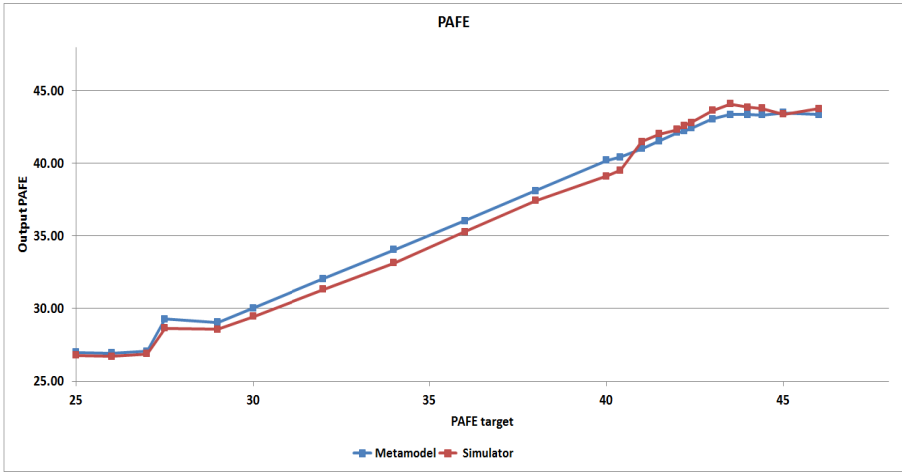


Fig. 4. Comparison of results obtained from the simulator & progressive meta-modeling for PAFE

The average percentage error for the problem is 4.38% for CM and 0.51% for PAFE. This is a significant improvement over the normal meta-modeling technique, where the percentage errors for CM and PAFE were found to be 6.82% and 2.74%, respectively.

Another interesting aspect of the study is that we are able to obtain portfolio solutions for a large PAFE value of 45, which are difficult to achieve due to highly constrained search space. Our progressive meta-modeling methodology allows us to concentrate near the optimal region of the search space and get more useful solutions to get evaluated exactly, thereby providing better objective and constraint value information for meta-modeling near the optimal region.

7 Conclusions

We have developed a methodology for meta-modeling and optimization of problems having mixed-variables. The results obtained using the meta-modeling techniques have been found to be satisfactory for the given problem. The linear or the exponential strategy has worked well for data sampling of such problems and may be used to generate adequate data samples without resorting to full-factorial sampling. The padding technique can take care of discreteness and very well work with the existing Kriging methods for meta-modeling objective and constraint functions for a varying-dimension problems with high accuracy. The proposed genetic algorithm for optimization (VDGA) has been capable of optimizing the objective functions subject to inequality constraints effectively. Either or both the objective function and the constraints can be approximated by using the trained meta-model instead of the actual simulation. We have also suggested a progressive meta-modeling method to handle large-sized problems, where the points used to create the meta-model have been introduced judiciously and in an iterative manner. The proposed methods are pragmatic and are ready to be applied to more real-world optimization problems.

References

1. Box, G.E.P., Hunter, W.G., Hunter, J.S.: *Statistics for experimenters*. John Wiley & Sons, New York (1978)
2. Chapra, S.C., Canale, R.: *Numerical Methods for Engineers*, 5th edn. McGraw-Hill, Inc., New York (2006)
3. Cheng, B., Titterton, D.M.: Neural networks: a review from a statistical perspective. *Statistical Science* 9, 2–30 (1994)
4. Deb, K.: An efficient constraint handling method for genetic algorithms. *Comput. Methods Appl. Mech. Engrg.* 186, 311–338 (2000)
5. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. *Complex Systems* 9, 115–148 (1995)
6. Krige, D.G.: A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa* 52(6), 119–139 (1951)
7. Lophaven, S., Nielsen, H., Søndergaard, J.D.: *A matlab kriging tool- box version 2.0.*, Technical Report IMM-REP-2002-12, Informatics and Mathematical Modelling, Technical University of Denmark (2002)
8. Meckesheimer, M., Barton, R.R., Simpson, T.W., Limayem, F., Yannou, B.: Meta-modeling of combined discrete/continuous responses. *AIAA Journal* 39, 1950–1959 (2001)
9. Nain, P.K.S., Deb, K.: Computationally effective search and optimization procedure using coarse to fine approximations. In: *Proceedings of the Congress on Evolutionary Computation (CEC 2003)*, Canberra, Australia, pp. 2081–2088 (2003)
10. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. *Statistical Science* 4, 409–435 (1989)
11. Smith, M.: *Neural networks for statistical modeling*. Von Nostrand Reinhold, New York (1993)
12. Srivastava, S.: *Algorithms for Meta-modeling and Optimization for Mixed-variables*. Master's Thesis. Indian Institute of Technology Kanpur, Kanpur, UP, India (2011)
13. Jin, R., Chen, W., Simpson, T.W.: Comparative studies of meta-modeling techniques under multiple modeling criteria. *Structural and Multidisciplinary Optimization* 23, 1–13 (2000)
14. Simpson, T.W., Mauery, T.M., Korte, J.J., Mistree, F.: Comparison of response surface and kriging models for multidisciplinary design optimization. In: *7th AIAA/USAF/NASA/ISSMO Symp. on Multidisciplinary Analysis & Optimization*, vol. 1, pp. 381–391 (1998)
15. Yang, R.J., Gu, L., Liaw, L., Gearhart, C., Tho, C.H., Liu, X., Wang, B.P.: Approximations for safety optimization of large systems. In: *ASME Design Automation Conf.*, pp. DETC-00/DAC-14245 (2000)
16. Meckesheimer, M., Barton, R.R., Simpson, T.W., Limayem, F., Yannou, B.: Meta-modeling of combined discrete/continuous responses. *AIAA Journal* 39, 1950–1959 (2001)
17. Ryerkerk, M., Averill, R., Deb, K., Goodman, E.: Optimization for Variable-Size Problems Using Genetic Algorithms. In: *Proceedings of the 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Indianapolis, USA, pp. 2012–5569. AIAA, Reston (2012)

A General Variable Neighborhood Search Algorithm for the No-Idle Permutation Flowshop Scheduling Problem

M. Fatih Tasgetiren¹, Ozge Buyukdagli¹,
Quan-Ke Pan², and Ponnuthurai Nagaratnam Suganthan³

¹Industrial Engineering Department, Yasar University, Izmir, Turkey
{fatih.tasgetiren, ozge.buyukdagli}@yasar.edu.tr

²College of Computer Science, Liaocheng University,
Liaocheng, Shandong, P.R. China
panquanke@gmail.com

³School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore
epnsugan@ntu.edu.sg

Abstract. In this study, a general variable neighborhood search (GVNS) is presented to solve no-idle permutation flowshop scheduling problem (NIPFS), where idle times are not allowed on machines. GVNS is a metaheuristic, where inner loop operates a variable neighborhood descend (VND) algorithm whereas the outer loop carries out some perturbations on the current solution. We employ a simple insert and swap moves in the outer loop whereas iterated greedy (IG) and iterated local search (ILS) algorithms are employed in the VND as neighborhood structures. The results of the GVNS algorithm are compared to those generated by the variable iterated greedy algorithm with differential evolution (vIG_DE). The performance of the proposed algorithm is tested on the Ruben Ruiz' benchmark suite that is presented in <http://soa.iti.es/r Ruiz>. Computational results showed that the GVNS algorithm further improved 85 out of 250 best solutions found so far in the literature.

Keywords: no-idle permutation flowshop scheduling problem, general variable neighborhood search, heuristic optimization, metaheuristics.

1 Introduction

A flowshop is a commonly used production system in manufacturing industries. Generally, in manufacturing environments, the jobs should go through different processes till the end items are obtained. If the route of each job is different, then this environment is referred as job shop. The production environment with all jobs have the same route is called flowshop. Scheduling of a flowshop has an essential role in competitive environments; therefore this problem has been one of the most attractive subjects for researchers.

In a flowshop, there is more than one machine and each job must be processed on each of the machines. Each job has the same ordering of machines for its process sequence. Each job can be processed on one machine at a time, and each machine can process only one job at a time. For the permutation flowshop, the processing

sequences of the jobs are the same on each machine. In other words, jobs have a permutation and therefore, once a permutation is fixed for all jobs on the first machine, this permutation is maintained for all other machines. If one job is at the i^{th} position on machine 1, then this job will be at the i^{th} position on all the machines.

In order to measure the performance of scheduling in a flowshop, there are several criteria such as, makespan or due-date based performance measures. Makespan criterion, without any doubt, the most widely used performance measure in the literature.

In this study, a variant of permutation flowshop scheduling problem (PFSP), where no-idle times are allowed on machines, is considered with the makespan criterion, too. The no-idle constraint has an important role in scheduling environment, where expensive machinery is employed. Idling machines in such environments is not cost-effective. Another situation that production environment desires to have no-idle times in schedule, is when high setup time or costs exist so that shutting down the machines after initial setup is not wanted. In no-idle permutation flowshop scheduling (NIPFS) problem, each machine must process each job without any interruption from the beginning of the first job to the completion of the last job. In order to meet this constraint, delays may occur in the processing of the first job on any machine.

$F_m/prmu, no - idle/C_{max}$ is a well-known notation of the m-machine NIPFS problem where the makespan is minimized. In [2], it was shown that $F_3/prmu, no - idle/C_{max}$ is an NP-hard problem. Although it has a great importance in both theory and practical applications, it has not attracted much attention in the literature by the researchers. In [1], an exact algorithm to solve $F_2/prmu, no - idle/C_{max}$ optimally is presented. The first time, the problem is studied with the makespan criterion in [13]. In [14], several heuristic approaches were examined for the general m-machine no-idle PFSP with the makespan criterion.

Recently, heuristic approaches have attracted increasing attention by many researchers. A heuristic, based on the traveling salesman problem (TSP), for the the $F_3/prmu, no - idle/C_{max}$ was represented in [3]. In [6], it was presented an adaptation of the NEH heuristic [9] for the NIPFS problem and also studied the interactions between the no-idle and no-wait flowshops. In [11], an IG algorithm for the NIPFS problem with the makespan criterion was presented and examined the performance against the existing algorithms. In [12], it was presented a discrete artificial bee colony algorithm to solve the no-idle permutation flowshop scheduling problem with the total tardiness criterion whereas in [15], a variable iterated greedy algorithm was proposed, where the standard DE algorithm was modified and applied in such a way that the probability to apply IG algorithm to the specific individual in the target population and the destruction size of IG were optimized with a standard DE algorithm. In [7], a GVNS algorithm was proposed with very good results for the single machine scheduling problem to minimize the total weighted tardiness with the sequence dependent setup times. Therefore, inspiring from [7], in this study, a GVNS algorithm is proposed to solve the NIPFS problem with makespan criterion and compared the results with the algorithm that is proposed in [15].

The rest of the paper is organized as follows. In Section 2, NIPFS problem is defined. Details of GVNS algorithm are given in Section 3. Computational results are given in in Section 4. Finally, conclusions are given in Section 5.

2 No-Idle Permutation Flowshop Scheduling Problem

No-idle permutation flowshop scheduling is required when the production environment desires to have no-idle times in production schedule because of the high costs or setup complexity of the system. In order to avoid the troubles in the production environment, the schedule must be done carefully while considering the all systems behavior.

There are n ($j = 1, 2, \dots, n$) jobs to be processed successively on m ($k = 1, 2, \dots, m$) machines with the same sequence on each machine. Associated with each job j and machine k , there is a processing time $p(j, k)$.

The assumptions for this problem are can be outlined as follows:

- Each machine can perform at most one job at any given time;
- Each job can be processed on at most one machine at any given time;
- Processing sequences of jobs are same on each machine;
- There cannot be idle times between the start of processing the first job to the completion of processing the last job on any machine.

We follow the formulation for the NIPFS problem with makespan criterion in [10]. This formulation consists of forward and backward pass calculation. The complexities of the both formulations are the same. For this study, the forward pass calculation is selected to be used.

2.1 Forward Pass Calculation

Let the partial sequence of π , $\pi_j^E = \{\pi_1, \pi_2, \dots, \pi_j\}$ represent the sequence of jobs from the first job to the j^{th} job of sequence π where $1 < j < n$. The minimum difference, between the completion of processing the last job of π_j^E on machines k and $k + 1$ is denoted as $F(\pi_j^E, k, k + 1)$ and restricted by no-idle constraint. $F(\pi_j^E, k, k + 1)$ can be computed as:

$$F(\pi_1^E, k, k + 1) = p(\pi_1, k + 1) \quad k = 1, 2, \dots, m - 1 \quad (1)$$

$$F(\pi_j^E, k, k + 1) = \max\{F(\pi_{j-1}^E, k, k + 1) - p(\pi_j, k), 0\} + p(\pi_j, k + 1) \\ j = 2, 3, \dots, n \quad \text{and} \quad k = 1, 2, \dots, m - 1 \quad (2)$$

In formulation (1), difference between the completions of processing the last job of π_1^E which only includes one job, on machines k and $k + 1$ is given. Since there is only one job, $F(\pi_1^E, k, k + 1)$ can be calculated by considering processing time of that job on corresponding, $(k + 1)^{\text{th}}$ machine. In formulation (2), calculation of $F(\pi_j^E, k, k + 1)$ for $j > 1$ is represented. It can be calculated by not only considering processing time of j^{th} job on machine k , also adding the positive difference between the previous job's completion of processing on machines k and $k + 1$.

The completion time of last job, π_n on last machine m can be calculated as summation of $F(\pi_n^E, k, k + 1)$ value for all machines and the processing times of all previously processed jobs including π_n itself;

$$C(\pi_n, m) = C_{\max}(\pi_n^E) = \sum_{k=1}^{m-1} F(\pi_n^E, k, k+1) + \sum_{j=1}^n p(\pi_j, 1) \quad (3)$$

Then, for any job j , completion time on last machine m can be computed by subtracting the processing time of the next job, π_{j+1} from the completion time of π_{j+1} on machine m ;

$$C(\pi_j, m) = C(\pi_{j+1}, m) - p(\pi_{j+1}, m) \quad j = n-1, n-2, \dots, 1 \quad (4)$$

Makespan can also be defined as the maximum completion time of jobs on the last machine by using the no-idle constraint of this problem;

$$C_{\max}(\pi_n^E) = \max(C(\pi_1, m), C(\pi_2, m), \dots, C(\pi_n, m)) \quad (5)$$

3 General Variable Neighborhood Search Algorithm

Variable neighborhood search (VNS) is a common approach proposed by [8] to enhance the solution quality by systematic changes of neighborhoods. The basic steps of VNS algorithm can be summarized as follows:

Initially, a set of neighborhood structures, N_k is selected where $k = 1, 2, \dots, k_{max}$. Having a multi neighborhood structure makes VNS an effective algorithm since most local search heuristics employ only one structure, $k_{max} = 1$. Then the initial solution is generated either randomly or using heuristics, such as NEH heuristic [9]. First a shaking phase then local searches are applied to the solution. The stopping criteria can be selected as maximum CPU time allowed or maximum number of iterations. *Shaking* step of VNS algorithm provides randomness in search. If this step is eliminated from algorithm, variable neighborhood descent (VND) algorithm which is shown in Figure 1 is obtained.

```

Procedure VND( $\pi$ )
 $d_{max} = 2$ 
 $d = 1$ 
do{
     $\pi_1 = N'_d(\pi)$ 
    If  $f(\pi_1) < f(\pi)$ 
         $\pi = \pi_1$ 
         $d = 1$ 
    else
         $d = d + 1$ 
}while( $d \leq d_{max}$ )
return  $\pi$ 
endprocedure
    
```

Fig. 1. Variable Neighborhood Descent

An extended VNS algorithm called general variable neighborhood search (GVNS) that is proposed in [5]. It can be obtained by replacing the local search step of VNS with VND algorithm. In this study, a different version of the GVNS algorithm with insert and swap operations in outer loop and in the inner loop (VND), IG algorithm and iterated

local search (ILS) algorithm [4] is employed to solve the NIPFS problem. The pseudo code of the GVNS algorithm we employed in this study is given in Figure 2.

```

Procedure GVNS
 $\pi = NEH$ 
 $\pi_b = \pi$ 
 $k_{max} = 2$ 
 $k = 1$ 
do{
     $\pi_1 = N_k(\pi)$ 
     $\pi_2 = VND(\pi_1)$ 
    if  $f(\pi_2) < f(\pi)$ 
         $\pi = \pi_2$ 
         $k = 1$ 
    else
         $k = k + 1$ 
}while( $k \leq k_{max}$ )
return  $\pi_b$ 
endprocedure

```

Fig. 2. General Variable Neighborhood Search

GVNS algorithm starts with an initial solution that is generated by the NEH heuristic. Then, the shaking phase of GVNS in the outer loop is composed of two different operations; the *insert* and *swap* operations applies only one insert and one swap move, respectively, to shake the permutation, $N_1(\pi) = insert(\pi)$, $N_2(\pi) = swap(\pi)$ operations. For the inner loop in VND, the neighborhood structures are taken as two powerful local search algorithms such as iterated greedy and iterated local search. In other words, the neighborhood structures for VND are taken as $N_1'(\pi) = IG(\pi)$ and $N_2'(\pi) = ILS(\pi)$ algorithms, the pseudo codes of the algorithms are given in Figure 3 and Figure 4, respectively. For the details of the ILS algorithm, we refer to [4].

```

Procedure  $N_1'(\pi)$ 
 $\pi_1 = DestructConstruct(\pi)$ 
Flag = true
do{
     $\pi_2 = RIS(\pi_1)$ 
    If  $f(\pi_2) < f(\pi_1)$ 
         $\pi_1 = \pi_2$ 
        Flag = true
    else
        Flag = false
}while(Flag = true)
return  $\pi_1$ 
endprocedure

```

Fig. 3. N_1' neighborhood structure (IG)


```

Procedure  $N'_2(\pi)$ 
 $\pi_1 = \text{perturbation}(\pi)$ 
 $Flag = true$ 
do{
     $\pi_2 = RIS(\pi_1)$ 
    If  $f(\pi_2) < f(\pi_1)$ 
         $\pi_1 = \pi_2$ 
         $Flag = true$ 
    else
         $Flag = false$ 
}while( $Flag = true$ )
return  $\pi_1$ 
endprocedure

```

Fig. 4. N'_2 neighborhood structure (ILS)

In the VND loop, the IG algorithm takes the solution from the GVNS loop and applies a destruction and construction procedure. A local search is also applied. On the other hand, the ILS algorithm takes the solution from the IG algorithm if it fails and perturbs the solution by some insert moves randomly taken between 1 and 5. Then it applies the same local search.

In the GVNS algorithm, the key procedures are the NEH heuristic, destruction and construction procedure and the local search. NEH heuristic is proposed by [9] and has been recognized as the highest performing method for the permutation flowshop scheduling problem. For the details of the NEH heuristic, we refer to [15].

Destruction and Construction Procedure consists of two main steps; destruction step and construction step. In the destruction step, pre-determined parameter d many jobs are randomly chosen and removed from the current solution. Therefore, two partial solutions obtained; one consists of the removed jobs, in the order which they removed denoted as π^R , the other one is the remaining part of the initial solution with size $n - d$ and denoted as π^D . In the construction phase, basically all jobs in π^R is inserted into each position in π^D one by one, and finally the best permutation with the minimum makespan is selected.

We use the *Referenced Insertion Procedure (RIS)*, where as an initial step, a referenced permutation, π^R , is selected which is the best solution found so far in this study. Then, the first job of the π^R is determined and the position of this job is found in the current permutation π . This corresponding job is removed from π and inserted into all possible positions of permutation π . Next, second job of the π^R is found in the permutation π , removed and inserted into the positions of its own permutation. And the procedure goes on in this way, until all the jobs in the π^R is processed. The pseudo code of the RIS algorithm is given below, in Figure 5.

```

Procedure RIS( $\pi, \pi^R$ )
 $h = 1$ 
 $i = 1$ 
while( $i \leq n$ )do {
     $h = h(\text{mod})n$ 
     $\pi_1 = \text{remove job } \pi_k \text{ from } \pi, \text{ corresponding to job } \pi_h^R \text{ in the}$ 
        reference permutation  $\pi^R$ 
     $\pi_2 = \text{the best permutation obtained by inserting job } \pi_k \text{ in}$ 
        any possible position of } \pi_1
    if( $f(\pi_2) < f(\pi)$ )then {
         $\pi = \pi_2$ 
         $i = 1$ 
    } else {
         $h = h + 1$ 
    } endif
} endwhile
return  $\pi$ 
endprocedure

```

Fig. 5. Referenced Insertion Procedure

4 Computational Results

In this study, a newly modified GVNS algorithm is proposed to solve no-idle permutation flow shop scheduling problem. In order to test the performance of these

Table 1. Average relative percentage deviation of the algorithms for $t=30$

| Jobs | Machines | vIG_DE | GVNS |
|------|----------|--------|-------|
| 50 | 10 | 0.03 | 0.14 |
| | 20 | -0.04 | -0.04 |
| | 30 | -0.17 | -0.12 |
| | 40 | -0.41 | -0.41 |
| | 50 | -0.16 | -0.23 |
| 100 | 10 | 0.04 | 0.08 |
| | 20 | -0.09 | -0.03 |
| | 30 | -0.19 | -0.30 |
| | 40 | -0.65 | -0.95 |
| | 50 | -0.12 | -0.27 |
| 150 | 10 | 0.00 | 0.00 |
| | 20 | 0.05 | 0.02 |
| | 30 | -0.18 | -0.12 |
| | 40 | -0.07 | -0.25 |
| | 50 | -0.85 | -0.97 |
| 200 | 10 | 0.00 | -0.00 |
| | 20 | -0.07 | -0.05 |
| | 30 | -0.31 | -0.33 |
| | 40 | -0.26 | -0.61 |
| | 50 | -0.40 | -0.53 |

Table 1. (continued)

| Jobs | Machines | vIG_DE | GVNS |
|------|----------|--------|--------|
| 250 | 10 | -0.01 | -0.01 |
| | 20 | -0.03 | -0.01 |
| | 30 | -0.14 | -0.28 |
| | 40 | 0.02 | -0.14 |
| | 50 | -0.60 | -1.09 |
| 300 | 10 | 0.00 | 0.00 |
| | 20 | 0.00 | -0.05 |
| | 30 | 0.02 | -0.05 |
| | 40 | -0.34 | -0.32 |
| | 50 | -0.23 | -0.54 |
| 350 | 10 | 0.00 | 0.01 |
| | 20 | -0.01 | 0.00 |
| | 30 | -0.05 | -0.14 |
| | 40 | 0.08 | -0.17 |
| | 50 | -0.44 | -0.72 |
| 400 | 10 | 0.00 | -0.00 |
| | 20 | 0.04 | 0.00 |
| | 30 | 0.11 | -0.02 |
| | 40 | -0.04 | -0.16 |
| | 50 | -0.25 | -0.49 |
| 450 | 10 | 0.00 | 0.01 |
| | 20 | 0.00 | 0.04 |
| | 30 | -0.03 | -0.16 |
| | 40 | -0.09 | -0.23 |
| | 50 | -0.07 | -0.53 |
| 500 | 10 | 0.00 | 0.00 |
| | 20 | -0.04 | -0.04 |
| | 30 | 0.08 | -0.05 |
| | 40 | 0.10 | -0.17 |
| | 50 | -0.03 | -0.38 |
| Avg | | -0.12 | -0.213 |

Table 2. The new best known solutions obtained by GVNS algorithm

| Instance | GVNS | Instance | GVNS |
|-----------|-------|-----------|-------|
| 50_20_01 | 5646 | 250_50_03 | 26351 |
| 50_20_05 | 4874 | 250_50_04 | 25500 |
| 50_40_04 | 9495 | 250_50_05 | 27332 |
| 50_40_05 | 8904 | 300_20_03 | 20229 |
| 50_50_01 | 11584 | 300_30_01 | 26487 |
| 50_50_02 | 10857 | 300_30_03 | 24363 |
| 50_50_03 | 10873 | 300_30_05 | 22553 |
| 50_50_04 | 9890 | 300_40_03 | 25361 |
| 100_30_03 | 10549 | 300_40_04 | 27466 |
| 100_40_03 | 12411 | 300_50_01 | 31538 |
| 100_40_04 | 11778 | 300_50_02 | 29474 |

Table 2. (continued)

| Instance | GVNS | Instance | GVNS |
|-----------------|-------------|-----------------|-------------|
| 100_40_05 | 12902 | 300_50_03 | 30701 |
| 100_50_01 | 15998 | 350_10_01 | 19297 |
| 100_50_03 | 17571 | 350_30_03 | 27638 |
| 100_50_04 | 16626 | 350_40_01 | 29072 |
| 100_50_05 | 14746 | 350_40_02 | 29010 |
| 150_20_04 | 10903 | 350_40_05 | 29742 |
| 150_30_01 | 15482 | 350_50_02 | 32882 |
| 150_30_03 | 14650 | 350_50_03 | 34682 |
| 150_40_01 | 15935 | 350_50_04 | 36985 |
| 150_40_02 | 18075 | 400_20_04 | 25105 |
| 150_40_04 | 14555 | 400_40_01 | 37426 |
| 150_40_05 | 17234 | 400_40_03 | 34450 |
| 150_50_01 | 20298 | 400_40_05 | 32730 |
| 150_50_05 | 19241 | 400_50_03 | 37680 |
| 200_30_01 | 17034 | 400_50_04 | 40444 |
| 200_30_05 | 17970 | 400_50_05 | 35499 |
| 200_40_01 | 19909 | 450_30_02 | 32494 |
| 200_40_02 | 21708 | 450_30_03 | 31968 |
| 200_40_04 | 17420 | 450_30_04 | 33691 |
| 200_50_02 | 23429 | 450_30_05 | 33641 |
| 200_50_03 | 22296 | 450_40_01 | 39547 |
| 200_50_04 | 23929 | 450_40_05 | 35681 |
| 250_20_04 | 17639 | 450_50_01 | 37287 |
| 250_30_02 | 21824 | 500_30_02 | 39348 |
| 250_30_04 | 19744 | 500_30_04 | 33896 |
| 250_40_01 | 22780 | 500_30_05 | 38339 |
| 250_40_02 | 24098 | 500_40_03 | 40313 |
| 250_40_03 | 24337 | 500_40_05 | 36203 |
| 250_40_04 | 24741 | 500_50_01 | 46175 |
| 250_40_05 | 23468 | 500_50_04 | 42345 |
| 250_50_01 | 28548 | 500_50_05 | 43000 |
| 250_50_02 | 24276 | | |

algorithms, the benchmark suite presented in <http://soa.iti.es/r Ruiz> is used. This benchmark is designed for NIPFS problem with makespan criterion specifically, with the number of jobs $n = \{50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$ and the number of machines $m = \{10, 20, 30, 40, 50\}$. There are 50 combinations with different sizes and each combination has 5 different instances. Thus, there are 250 instances in total. $R=5$ runs were carried out for each instance for each algorithm. All results are compared with the best-known solutions presented in the website of Ruiz García, Rubén. In order to compare these results, an average relative percentage deviation is calculated for each combination by using the following equation;

$$\Delta_{\text{avg}} = \sum_{i=1}^R \left(\frac{(H_i - \text{Best}) \times 100}{\text{Best}} \right) / R \quad (6)$$

where H_i is the objective function value that is obtained in i^{th} run of each algorithm, Best is the best-known solution presented in <http://soa.iti.es/r Ruiz> and R is the number of runs. The stopping criterion is selected as a maximum run time of each algorithm which is defined as $T_{\max} = n(m/2) \times t$ milliseconds where the value of t can be taken as, $t = 30$ or $t = 60$ depending on the comparison case.

The proposed algorithm is coded in C++ and run on an Intel Core 2 Quad 2.66 GHz PC with 3.5 GB memory. For the Destruction and Construction procedure, the destruction size is fixed at $d = 8$. The perturbation strength of the ILS algorithm is taken randomly between 1 and 5. The computational results are given in Table 1 and compared to vIG_DE algorithm in [15]. In addition, the new best known solutions obtained by GVNS algorithm are given in Table 2.

5 Conclusion

In this study, we proposed a GVNS algorithm to solve no-idle permutation flowshop scheduling problem (NIPFS), where idle times are not allowed on machines. GVNS is a metaheuristic, where inner loop operates a variable neighborhood descend (VND) algorithm whereas the outer loop carries out some perturbations on the current solution. The results of the GVNS algorithm are compared to those generated by the variable iterated greedy algorithm with differential evolution (vIG_DE). The performance of the proposed algorithm is tested on the Ruben Ruiz' benchmark suite that is presented in <http://soa.iti.es/r Ruiz>. Computational results showed that the GVNS algorithm further improved 85 out of 250 best solutions found so far in the literature. Table 1 and Table 2 show the average relative percentage deviation of the algorithms and best so far solutions obtained by GVNS algorithm, respectively. Instances are indicated as x_y_z , where x is the number of jobs, y is the number of machines and z is the instance number.

References

1. Adiri, I., Pohoryles, D.: Flow-shop/no-idle or no-wait scheduling to minimize the sum of completion times. *Naval Research Logistics Quarterly* 29(3), 495–504 (1982)
2. Baptiste, P., Lee, K.H.: A branch and bound algorithm for the F|no-idle|Cmax. In: Lyon : Proceedings of the International Conference on Industrial Engineering and Production Management (IEPM), vol. 1, pp. 429–438 (1997)
3. Saadani, H., Guinet, A., Moalla, M.: Three stage no-idle flow-shops. *Computers and Industrial Engineering* 44, 425–434 (2003)
4. Lourenc, H.R., Martin, O., Stützle, T.: Iterated local search. G. K. F. Glover içinde. In: *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, Cilt 57, pp. s. 321–s. 353. Kluwer Academic Publishers, Norwell (2002)
5. Hansen, P., Mladenovic, N., Urošević, D.: Variable neighborhood search and the local search and local branching. *Computers & Operations Research* 33(10), 3034–3045 (2006)
6. Kalczynski, P.J., Kamburowski, J.: On no-wait and no-idle flow shops with makespan criterion. *European Journal of Operational Research* 178(3), 677–685 (2007)

7. Kirlik, G., Oguz, C.: A variable neighborhood search for minimizing total weighted tardiness with sequence dependent setup times on a single machine. *Computers & Operations Research* 39, 1506–1520 (2012)
8. Mladenovic, N., Hansen, P.: Variable neighborhood search. *Computers and Operations Research* 24, 1097–1100 (1997)
9. Nawaz, M., Ensore Jr., E.E., Ham, I.: A heuristic algorithm for the m-machine, njob flowshop sequencing problem. *OMEGA, The International Journal of Management Science* 11(1), 91–95 (1983)
10. Pan, Q.-K., Wang, L.: A novel differential evolution algorithm for no-idle permutation flowshop scheduling problems. *European Journal of Industrial Engineering* 2(3), 279–297 (2008)
11. Ruiz, R., Vallada, E., Fernández-Martínez, C.: Scheduling in Flowshops with No-Idle Machines. *Computational Intelligence in Flow Shop and Job Shop Scheduling Studies in Computational Intelligence* 230, 21–51 (2009)
12. Tasgetiren, M.F., Pan, Q., Suganthan, P.N., Oner, A.: A discrete artificial bee colony algorithm for the no-idle permutation flowshop scheduling problem with the total tardiness criterion. *Applied Mathematical Modelling* (2013)
13. Vachajitpan, P.: Job sequencing with continuous machine operation. *Computers and Industrial Engineering* 6(3), 255–259 (1982)
14. Woollam, C.R.: Flowshop with no idle machine time allowed. *Computers and Industrial Engineering* 10(1), 69–76 (1986)
15. Tasgetiren, M.F., Pan, Q., Suganthan, P.N., Buyukdagli, O.: A variable iterated greedy algorithm with differential evolution for the no-idle permutation flowshop scheduling problem. *Computers & Operations Research* 40, 1729–1743 (2013)

Design of Non-uniformly Weighted and Spaced Circular Antenna Arrays with Reduced Side Lobe Level and First Null Beamwidth Using Seeker Optimization Algorithm

Gopi Ram¹, Durbadal Mandal¹, Sakti Prasad Ghoshal², and Rajib Kar¹

¹ Department of Electronics and Communication Engineering,
National Institute of Technology, Durgapur, West Bengal, India

² Department of Electrical Engineering, National Institute of Technology, Durgapur,
West Bengal, India

{gopi203hardel,durbadal.bittu,
spghoshalnitdgp,rajibkarece}@gmail.com

Abstract. A design problem of non-uniform circular antenna arrays for the best optimal performance with the help of seeker optimization algorithm is dealt with in this paper. This problem is modeled as a simple optimization one. The algorithm is used to determine an optimum set of current excitation weights and antenna inter-element separations that provide radiation pattern with maximum reductions of side lobe level (SLL) and first null beamwidth (FNBW). Circular array antennas lying on x - y plane are assumed. The same algorithm is applied successively on circular arrays of 8, 10 and 12 elements. Various simulation results are presented. Performances of side lobe and FNBW are analyzed. Experimental results show considerable reductions of both SLL and FNBW with respect to the uniform case and those of some referred literature [3, 4] based on GA and PSO, respectively.

Keywords: Circular antenna arrays, Side lobe level, First null beamwidth, Seeker optimization algorithm.

1 Introduction

A lot of research works have been carried out in the past few decades on different antenna arrays in order to get improved radiation patterns. An Antenna Array is formed by assembly of radiating elements in an electrical or geometrical configuration. Total field of the Antenna Array is found by vector addition of the fields radiated by the individual elements [1]. This is important to reduce interference from the side lobes of the antenna. There are several parameters by varying which the radiation pattern can be modified [1-2]. A circular array has all its elements placed along the perimeter of a circle. A circular array is an array that has a configuration of very practical interest. Its applications span over radio detection finding, air and space navigation, underground propagation, radar sonar and many other systems. In this case, the antenna array design problem consists of finding current excitation weights

and antenna inter-element separations that provide a radiation pattern with maximum reductions of both SLL and FNBW. The classical optimization methods are not suitable for optimal design of complex, nonlinear, multimodal, non-differentiable antenna array design problem. So, evolutionary methods have been employed for the optimal design of antenna array design problem. Different evolutionary optimization techniques are as follows: Genetic Algorithm (GA) is inspired by the Darwin's "Survival of the Fittest" strategy [5-8]; Conventional PSO has mimicked the behaviour of bird flocking or fish schooling [9]. Several attempts have been taken towards the system identification problem with basic PSO and its modified versions [10-12]. The key advantage of PSO is its simplicity in computation and less number of steps is required in the algorithm. The major drawbacks of RGA and PSO are premature convergence and entrapment to suboptimal solution. In this paper, the capability of finding near optimal result in multidimensional search space using GA and PSO [3, 4] and the proposed Seeker optimization algorithm (SOA) [13-14] is individually investigated thoroughly for the optimal design of non-uniformly weighted and spaced circular antenna arrays with reduced SLL and FNBW. So, to enhance the performance of optimization algorithms in global search (exploration stage) as well as local search (exploitation stage), the authors suggest an alternative technique seeker optimization algorithm (SOA) to achieve much reduced SLL and FNBW for non-uniformly weighted and spaced circular antenna arrays. In this paper the performances of GA and PSO given in the literature [3, 4] and the proposed SOA are analysed to finally demonstrate the effectiveness and superior performance of SOA for achieving the global optimal solution in terms of reduced SLL and FNBW. SOA [13-14] is essentially a population based heuristic search algorithm. It is based on human understanding and searching capability for finding an optimum solution. In SOA, optimum solution is regarded as one which is searched out by a seeker population. The underlying concept of SOA is very easy to model and relatively easier than other optimization techniques prevailing in the literature. The highlighting characteristic features of this algorithm are the following:

- (a) The performance of SOA varies a little with its parameters
- (b) Search direction and step length are directly used in this algorithm to update the position,
- (c) Proportional selection rule is applied for the calculation of search direction, which can improve the population diversity so as to boost the global searching ability and decrease the number of control parameters making it simpler to implement, and
- (d) Fuzzy reasoning is used to generate the step length because the uncertain reasoning of human searching could be the best described by natural linguistic variables, and a simple *if-else* control rule.

The present work focuses on the performance of the SOA for finding the optimal design of non-uniformly weighted and spaced circular antenna arrays.

2 Design Equation

Fig. 1 assumes the geometry of a circular array having a radius 'a' and N isotropic sources laid on x-y plane and scanning at point P in the far field. The elements in the non-uniform circular antenna array are taken to be isotropic sources, so the radiation pattern of this array can be described by its array factor. In the x-y plane, the array factor for the circular array shown in Fig.1 is given by (1) [1].

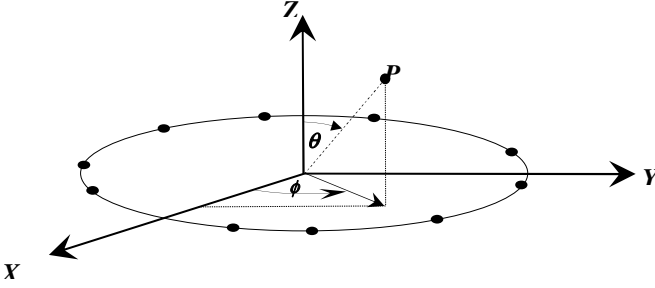


Fig. 1. Geometry of non-uniform circular array laid on the x-y plane with N isotropic elements scanning at a point P in the far field

$$AF(\varphi, I, d) = \sum_{n=1}^N I_n e^{j(ka \cos(\varphi - \varphi_n) + \alpha_n)} \quad (1)$$

where

$$ka = 2\pi a / \lambda = \sum_{i=1}^N d_i ; \quad (2)$$

$$\varphi_n = (2\pi / ka) \sum_{i=1}^n d_i ; \quad (3)$$

$k = 2\pi / \lambda$, λ being the wavelength of operation;

θ = Azimuth angle;

φ = Elevation angle;

φ_n = Angular location of nth element along the x-y plane;

$$\alpha_n = -ka \cos(\varphi_0 - \varphi_n) \quad (4)$$

In this case the array factor can be written as

$$AF(\varphi_n, I, d) = \sum_{n=1}^N I_n \exp\{jka[\cos(\varphi - \varphi_n) - \cos(\varphi_0 - \varphi_n)]\} \quad (5)$$

where $I = [I_1, I_2, \dots, I_N]$; I_n represents the excitation weight of the nth element of the array; $d = [d_1, d_2, \dots, d_N]$; d_n represents the distance from nth element to

$(n+1)^{\text{th}}$ element; φ_0 is the angle where the global maximum is attained in $\varphi = [-\pi, \pi]$. In our design problem φ_0 is chosen to be 0, i.e., φ_0 is the maximum radiation angle. The design goal in the paper is to find the optimum set of values of current excitation weights, I_n and inter-element spacing, d_n in order to get optimal reduction of both SLL and FNBW in the radiation pattern in the desired direction φ . I_n of each element and d_n are used to change the antenna radiation pattern.

After defining the array factor, the next step in the design process is to formulate the cost function which is to be minimized. The cost function (f_1) may be written as $f_1 = W_1 \times |AF(\varphi_{mst}, I_n)| / |AF(\varphi_0, I_n)| + W_2 \times (FNBW_{computed} - FNBW(I_n = 1))$ (6) where $FNBW$ is an abbreviated form of first null beamwidth or in simple terms angular width between first nulls on either side of the main beam. Thus, $FNBW_{computed}$ and $FNBW(I_n = 1)$ basically refer to the computed first null beamwidths in radian for the non-uniform case and for the uniform case, respectively. The second term in (6) is computed only if $FNBW_{computed} < FNBW(I_n = 1)$ and corresponding solution set of I_n and d_n is retained in the active population (otherwise discarded). Further, W_1 and W_2 are the weighting factors. φ_{mst} is the angle where maximum side lobe $AF(\varphi_{mst}, I_n)$ is attained on either side of the main beam. The weights W_1 and W_2 are chosen in such a way that optimization of SLL remains more dominant than optimization of $FNBW$ and f_1 never becomes negative. Minimization of f_1 means maximum reduction of SLL and lesser $FNBW_{computed}$ as compared to $FNBW(I_n = 1)$. The SOA employed for optimizing I_n and d_n resulting in the minimization of f_1 and hence both SLL and $FNBW$ is described in the next section.

3 Seeker Optimization Algorithm

Seeker Optimization Algorithm (SOA) [13-14] is a population-based heuristic search algorithm. It regards the optimization process as an optimal solution obtained by a seeker population. Each individual of this population is called a seeker. The total population is randomly categorized into three subpopulations. These subpopulations search over several different domains of the search space. All the seekers in the same subpopulation constitute a neighbourhood. This neighbourhood represents the social component for the social sharing of information.

3.1 Steps of Seeker Optimization Algorithm

3.1.1 Calculation of Search Direction, $d_{ij}(t)$

In the SOA, a search direction $d_{ij}(t)$ and a step length $\alpha_{ij}(t)$ are computed separately for each i th seeker on each j th variable at each time step t , where $\alpha_{ij}(t) \geq 0$ and $d_{ij}(t) \in \{-1, 0, 1\}$. Here, i represents the population number and j represents the optimizing variable number.

It is the natural tendency of the swarms to reciprocate in a cooperative manner while executing their needs and goals. Normally, there are two extreme types of cooperative behaviour prevailing in swarm dynamics. One, egotistic, is entirely pro-self and another, altruistic, is entirely pro-group. Every seeker/swarm, as a single sophisticated agent, is uniformly egotistic. He believes that he should go toward his historical best position according to his own judgment. This attitude of i th seeker may be modelled by an empirical direction vector $\vec{d}_{i, ego}(t)$ as in (7).

$$\vec{d}_{i, ego}(t) = \text{sign}(\vec{p}_{i, best}(t) - \vec{x}_i(t)) \quad (7)$$

In (7), $\text{sign}(\cdot)$ is a signum function on each variable of the input vector. On the other hand, in altruistic behaviour, seekers want to communicate with each other, cooperate explicitly, and adjust their behaviours in response to the other seekers in the same neighbourhood region for achieving the desired goal. That means the seekers exhibit entirely pro-group behaviour. The population then exhibits a self-organized aggregation behaviour of which the positive feedback usually takes the form of attraction toward a given signal source. Two optional altruistic directions may be modelled as in (8)-(9).

$$\vec{d}_{i, alt1}(t) = \text{sign}(\vec{g}_{best}(t) - \vec{x}_i(t)) \quad (8)$$

$$\vec{d}_{i, alt2}(t) = \text{sign}(\vec{l}_{best}(t) - \vec{x}_i(t)) \quad (9)$$

In (8)-(9), $\vec{g}_{best}(t)$ represents neighbours' historical best position, $\vec{l}_{best}(t)$ means neighbours' current best position.

Moreover, seekers enjoy the properties of pro-activeness; seekers do not simply act in response to their environment; they are able to exhibit goal-directed behaviour. In addition, the future behaviour can be predicted and guided by the past behaviour. As a result, the seeker may be pro-active to change his search direction and exhibit goal-directed behaviour according to his past behaviour. Hence, each seeker is associated with an empirical direction called as pro-activeness direction as given in (10).

$$\vec{d}_{i, pro}(t) = \text{sign}(\vec{x}_i(t_1) - \vec{x}_i(t_2)) \quad (10)$$

In (10), $t_1, t_2 \in \{t, t-1, t-2\}$ and it is assumed that $\vec{x}_i(t_1)$ is better than $\vec{x}_i(t_2)$. Aforementioned four empirical directions as presented in (7)-(10) direct human being to take a rational decision in his search direction.

If the j th variable of the i th seeker goes towards the positive direction of the coordinate axis, $d_{ij}(t)$ is taken as +1. If the j th variable of the i th seeker goes towards the negative direction of the coordinate axis, $d_{ij}(t)$ is assumed as -1. The value of $d_{ij}(t)$ is assumed as 0 if the i th seeker stays at the current position. Every variable j of $\vec{d}_i(t)$ is selected by applying the following proportional selection rule as stated in (11).

$$d_{ij} = \begin{cases} 0, & \text{if } r_j \leq p_j^{(0)} \\ +1, & \text{if } p_j^{(0)} \leq r_j \leq p_j^{(0)} + p_j^{(+1)} \\ -1, & \text{if } p_j^{(0)} + p_j^{(+1)} < r_j \leq 1 \end{cases} \quad (11)$$

In (11), r_j is a uniform random number in $[0, 1]$, $p_j^{(m)}$ ($m \in \{0, +1, -1\}$) is the percent of the numbers of “ m ” from the set $\{d_{ij,ego}, d_{ij,alt1}, d_{ij,alt2}, d_{ij,pro}\}$ on each variable j of all the four empirical directions, i.e. $p_j^{(m)} = (\text{the number of } m) / 4$.

3.1.2 Calculation of Step Length, $\alpha_{ij}(t)$

Different optimization problems often have different ranges of fitness values. To design a fuzzy system to be applicable to a wide range of optimization problems, the fitness values of all the seekers are turned into the sequence numbers from 1 to S as the inputs of fuzzy reasoning. The linear membership function is used in the conditional part since the universe of discourse is a given set of numbers, i.e. 1, 2,, S . The expression is presented as in (12).

$$\mu_i = \mu_{\max} - \frac{S - I_i}{S - 1} (\mu_{\max} - \mu_{\min}) \quad (12)$$

In (12), I_i is the sequence number of $\vec{x}_i(t)$ after sorting the fitness values, μ_{\max} is the maximum membership degree value which is equal to or a little less than 1.0. Here, the value of μ_{\max} is taken as 0.95.

A fuzzy system works on the principle of control rule as “If {the conditional part}, then {the action part}”. Bell membership function $\mu(x) = e^{-x^2/2\delta^2}$ is well utilized in the literature to represent the action part. The membership degree values of the input variables beyond $[-3\delta, +3\delta]$ are less than 0.0111 ($\mu(\pm 3\delta) = 0.0111$), and the elements beyond $[-3\delta, +3\delta]$ in the universe of discourse can be neglected for a linguistic atom. Thus, the minimum value $\mu_{\min} = 0.0111$ is set. Moreover, the parameter, of the Bell membership function is determined by (13).

$$\vec{\delta} = \omega \times abs(\vec{x}_{best} - \vec{x}_{rand}) \quad (13)$$

In (13), the absolute value of the input vector as the corresponding output vector is represented by the symbol $abs(\cdot)$. The parameter ω is used to decrease the step length with increasing time step so as to gradually improve the search precision. In the present experiments, ω is linearly decreased from 0.9 to 0.1 during a run. The \vec{x}_{best} and \vec{x}_{rand} are the best seeker and a randomly selected seeker, respectively, from the same subpopulation to which the i th seeker belongs. It is to be noted here that \vec{x}_{rand} is different from \vec{x}_{best} and $\vec{\delta}$ is shared by all the seekers in the same subpopulation.

In order to introduce the randomness in each variable and to improve local search capability, the following equation is introduced to convert μ_i into a vector $\vec{\mu}_i$ with elements as given by (14).

$$\mu_{ij} = RAND(\mu_i, 1) \quad (14)$$

In (14), $RAND(\mu_i, 1)$ returns a uniformly random real number within $[\mu_i, 1]$. Equation (15) denotes the action part of the fuzzy reasoning and gives the step length (α_{ij}) for every variable j .

$$\alpha_{ij} = \delta_j \sqrt{-\ln(\mu_{ij})} \quad (15)$$

3.1.2.1 Updating of Seekers' Positions

In a population of size S , for each seeker i ($1 \leq i \leq S$), the position update on each variable j is given by (16).

$$x_{ij}(t+1) = x_{ij}(t) + \alpha_{ij}(t) \times d_{ij}(t) \quad (16)$$

3.1.3 Subpopulations Learn from Each Other

Each subpopulation is searching for the optimal solution using its own information. It hints that the subpopulation may trap into local optima yielding a premature convergence. Subpopulations must learn from each other about the optimum information so far they have acquired in their respective domain. Thus, the position of the worst seeker of each subpopulation is combined with the best one in each of the other subpopulations using the following binomial crossover operator as expressed in (17).

$$x_{k_n j, worst} = \begin{cases} x_{l_j, best} , & \text{if } rand_j \leq 0.5 \\ x_{k_n j, worst} , & \text{else} \end{cases} \quad (17)$$

In (17), $rand_j$ is a uniformly random real number within $[0, 1]$; $x_{k_n j, worst}$ is denoted as the j th variable of the n th worst position in the k th subpopulation; $x_{l_j, best}$ is the j th variable of the best position in the l th subpopulation. Here, $n, k, l = 1, 2, \dots, K - 1$ and $k \neq l$. In order to increase the diversity in the population, good information acquired by each subpopulation is shared among the other subpopulations.

4 Experimental Results

The SOA described in the previous section is implemented to study the behavior of the radiation pattern for non-uniform circular antenna arrays. In this case radiation pattern of the circular array with main lobe steered to $\varphi_0 = 0$ degrees is considered. This section gives simulation results of both reduced SLL and FNBW in the radiation patterns. Three non-circular antenna arrays each having 8, 10, 12 elements are assumed. The SOA is executed for 400 iterations for each array. The population size has been fixed at 120. Table-I shows the best chosen parameters for the SOA.

Figures 2-4 show comparisons between the radiation patterns for a uniform circular antenna array ($d = \lambda / 2$) and the non-uniform circular antenna array optimized by the SOA. In the case of uniform circular array, inter-element spacing d is the arc distance between every pair of adjacent elements arranged in a circle, and radius $a = N \cdot \lambda / (4\pi)$.

Fig.2 indicates that the uniform circular array has a radiation pattern with -4.17 dB SLL for $N=8$, when excitation weights are of equal unity amplitudes. All side lobes are suppressed to a level -11.08 dB, and FNBW is reduced to 65.16 degrees as a result of the optimization by the SOA whereas [3-4] show SLLs of -9.811 dB, -10.799 dB and FNBW of 70.27 degrees, respectively, as shown in Table 2. Fig.2 shows the comparative radiation patterns as obtained using the results of Table 2 for $N=8$.

Fig 3 illustrates the case for $N=10$. For this value of N , the SOA provides a radiation pattern with minimum SLL of -12.83 dB and FNBW of 42.48 degrees as compared to SLLs of -9.811 dB, -12.307 dB and FNBW of 55.85 degrees obtained, respectively, in [3-4], shown in Table II.

Lastly, Fig 4 illustrates the case for $N=12$. For this value of N , as shown in Table 2, the SOA provides a radiation pattern with -13.77 dB SLL, i.e. the lesser minimum SLL, and reduced FNBW of 36 degrees as compared to SLLs of -11.88 dB, -13.670 dB and FNBW of 46.26 degrees, respectively, in [3-4]. Table 2 shows the results of element distribution and excitation distribution for each case. The same table illustrates that as the number of antenna elements N increases, the SLL reduction for non-uniform circular antenna array increases. It should be noted that the size of the circular array obtained is slightly larger than that obtained in [3-4] because in the present work, inter-element distance is maintaining the minimum distance ($\lambda / 2 < d < \lambda$) to avoid the mutual coupling effect.

5 Convergence Profile of SOA

The minimum f_1 values are recorded against number of iteration cycles to get the convergence profile of cost function obtained by the SOA for each optimal array design. Fig. 5 portrays the convergence profile of minimum f_1 for the circular array set having 12 elements. The programming was written in MATLAB language using MATLAB 7.5 on dual core (TM) processor, 2.88 GHz with 1 GB RAM.

Table 1. SOA parameters

| Parameters | SOA |
|--------------------------|------|
| Population Size | 120 |
| Maximum iteration cycles | 400 |
| Δ | 0.02 |
| k_1 | 1.5 |
| k_2 | 2.0 |
| k_3 | 2.0 |

Table 2. Example of element distributions and the resulting excitation distribution and inter-element separations for non-uniform circular antenna arrays obtained by the SOA for different numbers of antenna elements

| No. of Element s | SLL (dB) | SLL (dB) in [3], [4] | FNBW (deg) | FNBW (deg) in [3], [4] | $[I_1, I_2, I_3, I_4, \dots, I_N]$ $[d_1, d_2, d_3, d_4, \dots, d_N]$ in λ_s | Aperture in (λ) |
|---------------------|----------|----------------------|------------|------------------------|--|---------------------------|
| 8 | -11.08 | -9.811, -10.799 | 65.16 | 70.27, 70.27 | 0.3802 0.3721 0.9383 | 5.7928 |
| | | | | | 0.8865 0.2687 0.4840 | |
| | | | | | 0.2761 0.9208; | |
| | | | | | 0.9487 0.8199 0.8582 | |
| | | | | | 0.5675 0.7003 0.8517 | |
| | | | | | 0.5390 0.5075 | |
| | | | | | 0.4472 0.2924 0.3356 | |
| | | | | | 0.4214 0.5818 0.3783 | |
| 10 | -12.83 | -9.811, -12.307 | 42.48 | 55.85, 55.85 | 0.1824 0.1508 0.3818 | 7.9234 |
| | | | | | 0.5722 ; | |
| | | | | | 0.6047 0.9782 0.7718 | |
| | | | | | 0.9409 0.6405 0.9776 | |
| | | | | | 0.7347 0.5419 0.7819 | |
| | | | | | 0.9512 | |
| | | | | | 0.5169 0.4519 0.2006 | |
| | | | | | 0.5273 0.5617 0.8967 | |
| 12 | -13.77 | -11.83, -13.670 | 36.00 | 46.26, 46.26 | 0.6691 0.5912 0.2432 | 9.2770 |
| | | | | | 0.5839 0.6416 | |
| | | | | | 0.8285 ; | |
| | | | | | 0.8564 0.8305 0.6790 | |
| | | | | | 0.5796 0.8978 0.7911 | |
| | | | | | 0.5337 0.9848 0.7859 | |
| | | | | | 0.8148 0.9462 | |
| | | | | | 0.5772 | |

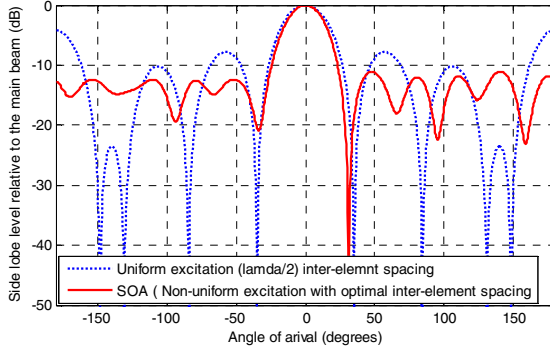


Fig. 2. Best array pattern found by the SOA for the case of 8-element non-uniform circular array

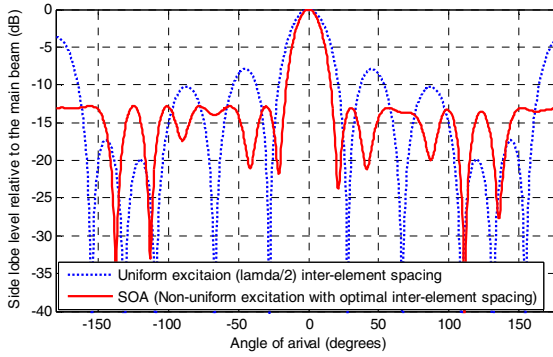


Fig. 3. Best array pattern found by the SOA for the 10-element non-uniform circular array

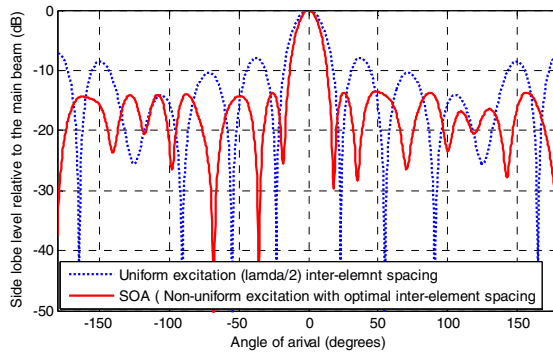


Fig. 4. Best array pattern found by the SOA for the 12-element non-uniform circular array

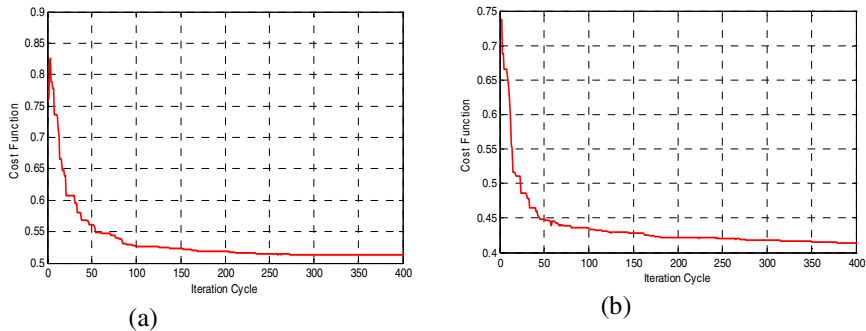


Fig. 5. Convergence profiles of SOA for the (a) 10-element and (b) 12-element non-uniform circular arrays

6 Conclusions

This paper illustrates how to model the optimal design of non-uniform circular antenna array for maximum reductions of side lobe level and first null beamwidth by a new approach named seeker optimization algorithm. The seeker optimization algorithm efficiently computes the parameters of non-uniform circular antenna array to generate the near optimal radiation pattern. Experimental results reveal that design of non-uniform circular antenna arrays using the same algorithm provides considerable reductions of side lobe level and first null beamwidth as compared to the uniform case and some published works as well. Also, array patterns obtained by the SOA are generally better than those presented in [3] and [4].

Future research will be aimed at dealing with other geometries and constraints. Many different areas of antenna design and analysis require a feasible and versatile procedure, being able to perform array synthesis by suitably tuning antenna characteristics and parameters. Seeker optimization algorithm has proved versatile and robust for the present work and earlier reported works and thus seems a good candidate to face the complex nonlinear problem of array antenna design.

Acknowledgment. We would like to thank SERB, Department of Science and Technology, Government of India (project no. SB/EMEQ-319/2013) for providing the fund which has been required to pursue this research.

References

1. Ballanis, C.A.: Antenna theory analysis and design, 2nd edn. John Wiley and Son's Inc., New York (1997)
2. Zainud Deen, S., Mady, E., Awadalla, K., Harsher, H.: Controlled Radiation Pattern of Circular Antenna Array. In: IEEE Antennas and Propagation Symp., pp. 3399–3402 (2006)

3. Panduro, M., Mendez, A., Dominguez, R., Romero, G.: Design of Non-uniform Antenna Arrays for Side Lobe Reduction Using the Method of Genetic Algorithm. *Int. J Electron. Communication (AEU)* 60, 713–717 (2006)
4. Sahib, M., Najjar, Y., Dib, N., Khodier, M.: Design of Non-uniform Circular Antenna Arrays Using the partial swarm optimization. *Journal of Electrical Engineering* 59(9), 216–220 (2008)
5. Haupt, R.L., Werner, D.H.: *Genetic Algorithms in Electromagnetics*. IEEE Press Wiley-Interscience (2007)
6. Marcano, D., Duran, F.: Synthesis of Antenna Arrays Using Genetic Algorithms. *IEEE Antennas Propagat. Magazine* 42(3) (June 2000)
7. Wang, L.L., Fang, D.G.: Synthesis of nonuniformly spaced arrays using genetic algorithm. In: *Asia-Pacific Conference on Environmental Electromagnetics*, pp. 302–305 (November 2003)
8. Eberhart, R., Shi, Y.: Comparison between Genetic Algorithms and Particle Swarm Optimization. In: *Proc. 7th Ann. Conf. on Evolutionary Computation*, San Diego (2000)
9. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: *Proc. IEEE Int. Conf. on Neural Network* (1995)
10. Mandal, D., Ghoshal, S.P., Bhattacharjee, A.K.: Determination of the Optimal Design of Three-Ring Concentric Circular Antenna Array Using Evolutionary Optimization Techniques. *International Journal of Recent Trends in Engineering* 2(5), 110–115 (2009)
11. Mandal, D., Ghoshal, S.P., Bhattacharjee, A.K.: A Novel Particle Swarm Optimization Based Optimal Design of Three-Ring Concentric Circular Antenna Array. In: *IEEE International Conference on Advances in Computing, Control, and Telecommunication Technologies, ACT 2009*, pp. 385–389 (2009)
12. Pathak, N., Mahanti, G.K., Singh, S.K., Mishra, J.K., Chakraborty, A.: Synthesis of Thinned Planar Circular Array Antennas Using Modified Particle Swarm Optimization. *Progress In Electromagnetics Research Letters* 12, 87–97 (2009)
13. Shaw, B., Mukherjee, V., Ghoshal, S.: Seeker optimisation algorithm: application to the solution of economic load dispatch problems. *Generation, Transmission & Distribution, IET* 5(1), 81–91 (2011)
14. Saha, S.K., Ghoshal, S.P., Kar, R., Mandal, D.: A novel seeker optimization algorithm: application to the design of linear phase FIR filter. *IET Signal Processing* (2012)

Covariance Matrix Adaptation Evolutionary Strategy for the Solution of Transformer Design Optimization Problem

Selvaraj Tamilselvi and Subramanian Baskar

Thiagarajar College of Engineering, Madurai - 625 015, Tamilnadu, India
{tsee, sbee}@tce.edu

Abstract. Transformer design (TD) is a complex multi-variable, non-linear, multi-objective and mixed-variable problem. This paper discusses the application of Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) for distribution TD, minimizing three objectives; purchase cost, total life-time cost and total loss individually. Two independent variables; voltage per turn and type of magnetic material are proposed to append with the usual TD variables, aiming at cost effective and energy efficient TD. Three case studies with three sets of TD vectors are implemented to demonstrate the superiority of CMA-ES and modified design variables (MDV), in terms of cost savings and loss reduction. Fourth case study depicts the accuracy, faster convergence and consistency of CMA-ES. Effectiveness of the proposed methodologies has been examined with a sample 400KVA 20/0.4KV transformer design. Simulation results show that CMA-ES with MDV provide the best solution on comparison with conventional TD procedure and, Branch and bound algorithm for TD optimization problem.

Keywords: Transformer design, CMA-ES, Magnetic material, Purchase cost, Total life-time cost, Volt per turn.

1 Introduction

From the outline of research papers in TD, efforts are focused on TD optimization problem [1]. TD optimization can be minimization of no-load loss [2], [3], minimization of load loss [4], maximization of efficiency [5], maximization of rated power [6], minimization of mass [6] or minimization of cost [5], [7]-[12], based on the objective functions.

A deterministic method of geometric programming is applied for the minimizing total mass of the transformer [6]. But this method requires mathematical model. Bacterial foraging algorithm (BFA) [5] and simulated annealing technique [7] have been adopted for minimizing main material cost (MMC) of transformer. However, when the search space grows exponentially, basic BFA would not be suitable and the simulated annealing technique finds difficult in extending itself to the multi-objective case.

Hybrid Finite Element Method (FEM) in combination with boundary element method (BEM), i.e. FEM-BEM [13], and heuristic solution [8] have been implemented for the minimization of transformer manufacturing cost (MC). But the disadvantage of numerical field analysis is mesh size and, that heuristic method is trial and error based. In all the above mentioned TD objectives such as minimization of total mass, MMC and MC, losses were not considered.

Branch and bound algorithm (BBA) tailored to a mixed integer non-linear programming, i.e. BBA-MINLP [10] and another widely used numerical field analysis technique in combination with BEM [12] have addressed the minimization of TLTC of transformer. These papers have overcome the above said limitation, by including losses in the objective function calculation. However, the magnetic material (MM) is not optimized, whereas fixed as MOH-0.27 [10]. To take into consideration the conflicting transformer design objectives, for example: minimizing purchase cost, and minimizing TLTC, core material should not be fixed. Only when, the type of magnetic material (TMM) is preferred as a decision variable, material to be used to build the core, for the respective design objective can be easily optimized from the variety of available core materials, satisfying various other TD constraints.

Generally, TD calculations require accessing several look-up tables' data for the evaluation of specific core loss at various flux densities, winding gradient, oil gradient, and heat transmission. Such complex analytical calculations interacting with graphical data are not handled accurately by the derivative based methods discussed above and thus the optimal solution is not guaranteed for the TD optimization problem solved by the analytical methods.

Apart from the deterministic methods, genetic algorithm (GA) and neural network are also employed for the TD optimization. GA has been applied for the minimization of MC, incorporating TMM as design variable [9], transformer cost plus running cost minimization [11]. Neural network technique has been applied for the no load loss minimization [2], [3]. A more recent approach to adapting the mutation covariance matrix is the CMA-ES proposed by Hansen. Its important property is invariance against the linear transformations in the continuous search space, when compared to other evolutionary algorithms. CMA-ES finds a global or near optimal minimum without using derivatives of the objective functions. This algorithm outperformed its competitors in CEC'05 benchmark optimization problems. Hence in this paper CMA-ES is applied for solving this complex TD optimization problem [14].

The main contributions of the paper are: (a) application of CMA-ES for TD optimization for the first time, assuring accuracy, consistency and convergence; (b) incorporation of TMM as one of the design variables for representing 10 different materials; (c) inclusion of variable, voltage per turn in place of low voltage (LV) turns; (d) optimization of three different objectives such as minimization of purchase cost, minimization of TLTC, and minimization of total loss, individually suggesting the designer a set of optimal transformers instead of single solution, so that he can choose which of them best fits the requirement of the customer and application under consideration; (e) comparison of simulation results with recent report [10] and conventional transformer design procedure (CTDP) [15], which is multiple design method or heuristic technique.

This paper is organized as follows: Section 2 describes the design of distribution transformer, section 3 presents CMA-ES algorithm, section 4 explains the CMA-ES based TD optimization, section 5 includes computational results and section 6 concludes the paper.

2 Design of Distribution Transformer

2.1 Preliminary

- Performance variables: These parameters list include transformer rating i.e. name plate details, design requirements on guaranteed no-load loss (P_{GNLL}), guaranteed load loss (P_{GLL}), guaranteed short circuit impedance (U_{KS}), minimum full load efficiency, maximum temperature rise, voltage regulation, tolerances for no-load loss and load loss constraints (e_1, e_2), etc.
- Core variables: The core data include mainly the core stacking factor, mass density, magnetization curve, and values of specific core loss for different maximum magnetic flux densities at 50Hz frequency for 10 different magnetic materials such as M3-0.27, M4-0.27, MOH-0.23, MOH-0.27, 23ZDKH90, 27ZDKH95, 23ZH90, 23ZH95, 23ZDMH85, 27ZDMH. Specific core loss data has been taken from Nippon steel catalogue [16].
- Conductor variables: This list include the resistivity of copper at the maximum specified temperature, type of internal and external winding, typical practical values for insulation of conductor, distance and insulation between windings and core, mass density of conductor, distance between two adjacent cores, maximum ambient and winding temperature, High Voltage (HV) taps, etc. Copper sheet is used for LV conductor and copper wire is used for HV conductor.
- Cost variables: This includes unit price of main materials in euro per kilogram such as i^{th} MM ($C_{FE,i=1,2..10}$) i.e. ($C_{FE,1}, C_{FE,2}, \dots, C_{FE,10}$), conductor (C_{COND}), sheet steel (C_{SS}), mineral oil (C_{OIL}), insulating paper (C_{INS}), duct strips (C_{DS}), and corrugated panel (C_{CORR}).

2.2 Mathematical Model

Mathematically, TD optimization problem can be stated in the form as:

$$\begin{aligned}
 &\text{Find } x = \{x_1, x_2, \dots, x_n\}, \quad \text{which} \\
 &\text{Minimize } z_p(x), \quad \text{for } p = 1, 2, \dots, l \\
 &\text{Subject to } h_j(x) \leq d_j, \quad \text{for } j = 1, 2, \dots, q \\
 &\quad \quad \quad x_t \geq 0, \quad \text{for } t = 1, 2, \dots, n \\
 &\quad \quad \quad x_{t,min} \leq x_t \leq x_{t,max}, \quad \text{for } t = 1, 2, \dots, n
 \end{aligned} \tag{1}$$

where z_p , p^{th} objective function and h_j , j^{th} inequality constraint are functions of 'n' decision variables x ; d_j , are constants; and, $x_{t,min}$ and $x_{t,max}$ are lower and upper limit on x , respectively.

Minimum Purchase Cost Design-Objective Function z_1 . This objective minimizes the total cost of transformer main materials (C_{TM}) including core, LV conductor, HV conductor, insulating paper, duct strips, oil, corrugated panel, and sheet steel using [10], [15]:

$$\text{Min, } z_1 = C_{TM} = \{C_{FE,i} \cdot G_{FE,i} + C_{COND} \cdot G_{LV} + C_{COND} \cdot G_{HV} + C_{INS} \cdot G_{INS} + C_{DS} \cdot G_{DS} + C_{OIL} \cdot G_{OIL} + C_{CORR} \cdot G_{CORR} + C_{SS} \cdot G_{SS}\} \quad (2)$$

where C_{TM} , $G_{FE,i=1,2,..10}$, G_{LV} , G_{HV} , G_{INS} , G_{DS} , G_{OIL} , G_{CORR} , and G_{SS} are total main materials cost of transformer in Euro, weight of the i^{th} MM, LV conductor, HV conductor, insulating paper, duct strips, oil, corrugated panel, and sheet steel in kg respectively.

Minimum TLTC Design-Objective Function z_2 . The cost optimal design of transformer has to minimize the sum of transformer cost and running cost. In this objective, transformer's selling price and losses are considered as transformer price. So material saving and energy saving are the two important aspects in minimization of TLTC. This objective minimizes TLTC of transformer (C_{TLT}), using [10], [15] :

$$\text{Min, } z_2 = C_{TLT} = \left\{ \frac{(C_{TM} + C_{rem} + C_{lab})}{(1 - S_m)} + AA \cdot P_{NLL} + BB \cdot P_{LL} \right\} \quad (3)$$

where, C_{TLT} , AA , BB , C_{rem} , C_{lab} , S_m , P_{NLL} and P_{LL} are total life time cost of transformer in euro, no-load loss cost rate in euro per watts, load loss cost rate in euro per watts, remaining materials cost in euro, labor cost in euro, sales margin, designed no-load loss in watts and designed load loss in watts respectively.

Minimum Total Loss Design-Objective Function z_3 . It minimizes the total loss of the transformer (T_{los}) using [8]: $\text{Min, } z_3 = T_{los} = \{P_{NLL} + P_{LL}\}$ (4)

Design Vector- dv . Design variables are collectively called as design vector. Modified Design Variables (MDV) are described in the following three sets of design vectors ($dv_{i=1,..3}$) to realize the TD case study, detailed in section 5.

$$dv_1 = \{x_1, x_2, \dots, x_6\} \quad (5)$$

$$dv_2 = \{x_2, x_3, \dots, x_7\} \quad (6)$$

$$dv_3 = \{x_2, x_3, \dots, x_8\} \quad (7)$$

where x_1 (integer variable) is LV turns; x_2 (integer variable) is width of the core leg in mm; x_3 (integer variable) is height of the core window in mm; x_4 (discrete variable) is maximum magnetic flux density in Tesla; x_5 (discrete variable) is current density in LV winding in A/mm^2 ; x_6 (discrete variable) is current density in HV winding in A/mm^2 ; x_7 (discrete variable) is voltage per turn in volts; x_8 (integer variable) is TMM.

Constraints of Performance Indices. Transformer performance must meet certain standards and rules, specified in IEC 60076-1, include,

- No-load loss constraint: $P_{NLL} \leq P_{GNLL} (1 + e_1)$ (8)
- Load loss constraint: $P_{LL} \leq P_{GLL} (1 + e_1)$ (9)
- Total loss constraint: $(P_{NLL} + P_{LL}) \leq (P_{GNLL} + P_{GLL})(1 + e_2)$ (10)
- Impedance voltage constraint: $U_{ks} (1 - e_2) \leq U_k \leq U_{ks} (1 + e_2)$ (11)

where, U_k is the designed short circuit impedance.

Constraints of Manufacturing Process Indices. These constraints are based on the transformer manufacturer specifications.

- Ratio of width of the core leg to the height of the core window should be less than or equal to unity; $x_2 \leq x_3$ (12)
- 2-multiplied core leg thickness (T_{cl}) must be between a minimum of half core leg width and a maximum of 90% core leg width; $0.5x_2 \leq 2T_{cl} \leq 0.9x_2$ (13)

Constraints of Material Performance

- Heat transfer constraint: Total heat produced by total loss of transformer must be smaller than the total heat dissipated by convection and radiation (H_{diss}) through cooling arrangement; $P_{NLL} + P_{LL} \leq H_{diss}$ (14)
- Magnetic flux density constraint: $x_{4,min} \leq x_4 \leq x_{4,max}$ (15)

3 CMA-ES Algorithm

A standard CMA-ES with weighted intermediate recombination, step size adaptation, and a combination of rank - μ update and rank-one update is considered in this paper [14]. The various processes involved in the algorithm are discussed below in steps.

Step 0: Initialization

Set parameters $\lambda, \mu, \mu_{cov}, d_\sigma, \mu_{cov}, c_\sigma, c_c, w_i, c_{cov}$ to their default values detailed below. Set evolution path $p_\sigma^{(0)} = 0, p_c^{(0)} = 0$ and covariance matrix $C^{(0)} = I$. Choose step size $\sigma^{(0)}$ and distribution mean $m^{(0)}$. Initialize generation count, $g = 0$.

Step 1: Sampling population

Set of search points $X_1^{(g+1)}, \dots, X_\lambda^{(g+1)}$ are generated by sampling the distribution, using the mean value $m^{(g)}$, covariance matrix $C^{(g)}$ and step size ($\sigma^{(g)}$). The basic equation for sampling the search points, for generation number $g = 0, 1, 2, \dots$ is given as,

$$X_k^{(g+1)} = N \left(m^{(g)}, (\sigma^{(g)})^2 C^{(g)} \right), \text{ for } k = 1, \dots, \lambda \quad (16)$$

where $X_k^{(g+1)}$ is the k^{th} offspring from generation $g + 1$; λ is the population size.

Step 2: Selection and recombination

The (λ) sampled points are ranked in order of ascending fitness and (μ) best are selected. The new mean ($m^{(g+1)}$) of all current population vectors is a weighted

average of (μ) selected vectors from the samples $X_1^{(g+1)}, \dots, X_\lambda^{(g+1)}$ with weight parameter, (w_i) and is updated using: $m^{(g+1)} = \sum_{i=1}^{\mu} w_i X_{i:\lambda}^{(g+1)}$ (17)

Step 3: Covariance matrix Adaptation

Rank- μ update of C is estimated as

$$C^{(g+1)} = (1 - c_{cov})C^{(g)} + c_{cov} \left\{ \sum_{i=1}^{\mu} w_i \left(\frac{X_{i:\lambda}^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \right) \left(\frac{X_{i:\lambda}^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \right)^T \right\} \quad (18)$$

Evolution path ($p_c^{(g)}$) is computed as

$$p_c^{(g+1)} = (1 - c_c)p_c^{(g)} + \sqrt{\mu_{eff}c_c(2 - c_c)} \left(\frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \right) \quad (19)$$

The rank-one update of covariance matrix is given by

$$C^{(g+1)} = (1 - c_{cov})C^{(g)} + c_{cov}p_c^{(g+1)}p_c^{(g+1)T} \quad (20)$$

The final CMA update formula for the covariance matrix ($C^{(g+1)}$) combines (18) and (20), with $\mu_{cov} \geq 1$, weighting between rank- μ and rank-one update:

$$C^{(g+1)} = (1 - c_{cov})C^{(g)} + \frac{c_{cov}}{\mu_{cov}} p_c^{(g+1)} p_c^{(g+1)T} + c_{cov} \left(1 - \frac{1}{\mu_{cov}} \right) \sum_{i=1}^{\mu} w_i \left(\frac{X_{i:\lambda}^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \right) \left(\frac{X_{i:\lambda}^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \right)^T \quad (21)$$

where $C^{(g)} = B^{(g)}(D^{(g)})^2(B^{(g)})^T$; $\mu_{eff} = 1/(\sum_{i=1}^{\mu} w_i^2)$ is the variance effective selection mass; $c_c = \frac{4}{(n+4)}$ learning ratio determines the cumulation step for the evolution path; $B^{(g)}$ is an orthogonal basis of eigenvectors; $D^{(g)}$, diagonal elements are the square roots of the eigen values of $C^{(g)}$; μ_{cov} is parameter for weighting between rank-one and rank- μ update and

$c_{cov} = \left\{ \frac{1}{\mu_{cov}} \frac{2}{(n+\sqrt{2})^2} + \left(1 - \frac{1}{\mu_{cov}} \right) \min \left(1, \frac{(2\mu_{cov}-1)}{(n+2)^2 + \mu_{cov}} \right) \right\}$ is the learning rate of covariance matrix update $C^{(g)}$.

Step 4: Step-Size control

Conjugate evolution path is $p_\sigma^{(g+1)} = (1 - c_\sigma)p_\sigma^{(g)} + \{\sqrt{(c_\sigma(2 - c_\sigma)\mu_{eff}}) B^{(g)}D^{(g)-1}B^{(g)T} \left(\frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \right)\}$ (22)

The step size ($\sigma^{(g)}$) is adapted with conjugate evolution path ($p_\sigma^{(g)}$) as $\sigma^{(g+1)} = \sigma^{(g)} \exp \left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|p_\sigma^{(g+1)}\|}{E\|N(0,I)\|} - 1 \right) \right)$ (23)

where I is Identity matrix; $N(0, I)$ is normal distribution with zero mean and unity covariance matrix. $E\|N(0, I)\|$ is expectation of the Euclidean norm of a $N(0, I)$ distributed random vector. $p_\sigma^{(0)} = 0$ initially; $c_\sigma = \frac{10}{(n+20)}$ is backward time horizon of evolution path; $d_\sigma = \max \left(1, \frac{3\mu_{cov}}{(n+10)} + c_\sigma \right)$ is a damping parameter for the step size.

4 CMA-ES Based TD Optimization

The step by step CMA-ES implementation for TD optimization is given below. CMA-ES minimize objective functions z_1 to z_3 , subject to the constraints (8) to (15). Set the objective function from equations (2)-(4), design vector dv from equations (5)-(7), TD variables of section 2.1, lower and upper bounds for dv and constraints.

- Step 1: Input $C^{(0)}$, $m^{(0)}$. Set CMA-ES parameters λ , μ , μ_{cov} , d_σ , μ_{cov} , c_σ , c_c , w_i , c_{cov} etc. Initialize $p_\sigma^{(0)} = 0$, $p_c^{(0)} = 0$.
- Step 2: Choose step size $\sigma^{(0)}$ as $0.25(x_{t,max} - x_{t,min})$ and maximum number of generations (g_{max}).
- Step 3: Initialize generation count, $g = 0$.
- Step 4: If termination criterion: $g = g_{max}$ is met, go to step 11. Else go to Step 5.
- Step 5: Generate (λ) candidate solutions by sampling gaussian multi-variate distribution with covariance matrix and standard deviation from (16).
- Step 6: Determine the fitness function, ($FF(x)$), using penalty parameter-less constraint handling, where in-feasible solutions are compared based on their constraint violation:

$$FF(x) = \begin{cases} z_p(x), & \text{for } p = 1,2..4, \text{ if feasible} \\ z^{max} + \sum_{j=1}^q \langle h_j(x) \rangle, & \text{otherwise} \end{cases} \quad (24)$$

where $FF(x)$ is the fitness function, (z^{max}) is the objective function value of the worst feasible solution in the current population. The fitness of the in-feasible solutions not only depends on the overall constraint violation, but also on the fitness of the worst feasible solution. However, the fitness of a feasible solution is always fixed and is equal to its objective function value ($z_p(x)$). If there is no feasible solution in the current population, (z^{max}) is zero.

- Step 7 : The (λ) sampled points are ranked in order of ascending fitness. Select (μ) best search points. Update the mean value ($m^{(g+1)}$) using (17) and search points.
- Step 8 : Update the covariance matrix ($C^{(g+1)}$) by (19) and (21).
- Step 9 : Update global step size ($\sigma^{(g+1)}$), using (22), and (23).
- Step 10 : Increment generation count, g ; Go to step 4.
- Step 11 : Stop the optimization process.

5 Computational Results

To demonstrate the effectiveness of the proposed MDV, a design example of 400KVA, 50Hz, 20/0.4 KV, 3 phase, shell type, wound core transformer with vector group, Dyn11 has been considered. The upper and lower bounds of the design variables, AA, BB, unit price of transformer materials, C_{rem} , C_{lab} , S_m , e_1 , e_2 are derived from [10]. As reference transformer, the transformer with loss category AB' according to CENELEC is selected, which means that $P_{GNLL} = 750W$, $P_{GLL} = 4600W$, and

$U_{ks} = 4\%$. Coding for TD optimization is developed using MATLAB 7.4 on Intel core, i3 processor Laptop, operating at 3.2 GHZ, with 3 GB RAM. Suitable modifications are incorporated for handling TD constraints in the coding of CMA-ES [17]. The population size and max number of function evaluations are fixed at 100 and 10,000 respectively.

5.1 Case Study 1

This case study proves the superiority of CMA-ES in TD optimization. CMA-ES minimize objective functions z_1 , and z_2 , subject to the constraints (8) to (15), by finding optimum values for the design vector, dv_1 [10] of equation(5) and results of output variables (ov) like C_{TM} , C_{TLT} , P_{LL} , P_{NLL} , T_{los} are given in Table 1. It compares the solution of CMA-ES with BBA-MINLP [10] and CTDP [15]. In Table 1, three techniques converged to three different solutions for both the objectives. In particular, the CMA-ES converges to the best result. CMA-ES has given cost savings of about 8.801% and 15.84%, for z_1 and, 3.306% and 2.298%, for z_2 , when compared with BBA-MINLP method and CTDP.

5.2 Case Study 2

The CMA-ES optimizes the design vector dv_2 , minimizing the objective functions z_1 to z_3 , subject to the constraints (8) to (15), and simulation results are tabulated in Table 2. This case study discusses the effectiveness of the addition of proposed TD variable x_7 voltage per turn in place of x_1 LV turns in the existing design vector dv_1 taken from [10]. Variable x_7 is varied at discrete levels in the interval [0 15] with step 0.01. In this case, an additional constraint is included that variable x_1 must be an integer. The MM used for building the core is fixed as MOH-0.27 like case study 1 [10]. Objective values optimized with x_7 are found lesser than the results of the case before adding x_7 , for all the objectives. Optimization results of CMA-ES with dv_2 for all the objective functions have yielded performance improvement of 1.21%, 0.823%, and 2.24%, respectively on comparison with the results of dv_1 .

5.3 Case Study 3

In this case study, CMA-ES optimizes the design vector dv_3 , for the objective functions z_1 to z_3 , subject to the constraints from (8) to (15), and the results are depicted in Table 2. In the TD problem, soundness of the TD variable x_8 , TMM is analyzed by adding it with the design vector dv_2 . Here, MM used is not common for all the objective functions, whereas TMM is also optimized, during the run. By optimizing this way, no-load losses and cost due to no load losses for different MM can be evaluated during the design phase, and the appropriate core material, out of 10 materials can be optimized for each design objective considered. Variable x_8 is defined as an integer, such that it varies from 0 to 10 for representing the 10 magnetic materials. In this

case, an additional constraint is included that variable x_1 LV turns must be an integer. From Table 2, it is explicit that the optimization results of dv_3 by CMA-ES are superior to the solutions of dv_2 for the objective functions z_2 , and z_3 . The design variable x_8 can impart appreciable change in the objective values, only when the core loss calculation is used during the fitness function evaluation. Since z_1 design does not rely on losses computation, TMM optimized in this case study 3 (material no: 4) is same as that of the MM fixed in case study 2, i.e. MOH – 0.27. On comparison with CMA-ES dv_2 , and CMA-ES dv_1 respectively, CMA-ES dv_3 has given cost savings of about 1.89%, 2.73%, for z_2 , and loss reduction of about 5.17%, 7.53% for z_3 . From the simulation results, it is evident that CMA-ES with proposed MDV is able to find the optimal solution for TD optimization problem, irrespective of the objective function.

5.4 Case Study 4

Due to the stochastic nature of CMA-ES, 25 independent runs are performed to prove the consistency in obtaining the optimal solutions and their statistical results such as mean variance and standard deviation (SD) are reported in this case study for all the three objective functions z_1 to z_3 , using dv_3 . The performance of CMA-ES is shown with respect to solution accuracy, and mean computation time (MCT) to reach the optimum in Table 3. For all the objective functions employed, the numerical results of CMA-ES given in Table 3 are found more satisfactory, in terms of performance, consistency, faster convergence for the TD optimization problem. Convergence plot for objective function z_2 with dv_3 is shown in Fig.1.

Table 1. Optimization Results of CMA-ES with dv_1

| dv_1 / ov | z_1 | | | z_2 | | |
|---------------------|--------------|------------------------|-------------|--------------|-----------------------|--------------|
| | CTDP [15] | BBA MINLP P [10] | CMA- ES | CTDP [15] | BBA- MINLP [10] | CMA- ES |
| x_1 | 17 | 18 | 18 | 17 | 20 | 17 |
| x_2 | 220 | 239 | 212 | 220 | 231 | 242 |
| x_3 | 245 | 248 | 241 | 245 | 299 | 242 |
| x_4 | 1.7 | 1.8 | 1.8 | 1.7 | 1.6 | 1.6 |
| x_5 | 3.0 | 3.0 | 3.6 | 3.0 | 3.0 | 3.0 |
| x_6 | 3.0 | 3.0 | 3.6 | 3.0 | 3.0 | 3.0 |
| C_{TM} | 4475 | 4203 | 3863 | 4475 | 4428 | 4617 |
| P_{LL} | 4148 | 4288 | 5012 | 4148 | 4613 | 4265 |
| P_{NLL} | 818 | 859 | 856 | 818 | 719 | 685 |
| C_{TLT} | 27199 | 27500 | 28790 | 27199 | 27467 | 26588 |

Table 2. Optimization Results of CMA-ES with dv_2 and dv_3

| dv $/ov$ | z_1 | | | z_2 | | | z_3 | | |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | with dv_1 | with dv_2 | with dv_3 | with dv_1 | with dv_2 | with dv_3 | with dv_1 | with dv_2 | with dv_3 |
| x_1 | 18 | ^k NA | ^k NA | 17 | ^k NA | ^k NA | 16 | ^k NA | ^k NA |
| x_2 | 212 | 224 | 224 | 242 | 245 | 245 | 232 | 220 | 219 |
| x_3 | 241 | 227 | 227 | 242 | 245 | 245 | 232 | 220 | 219 |
| x_4 | 1.8 | 1.8 | 1.8 | 1.6 | 1.6 | 1.7 | 1.6 | 1.6 | 1.75 |
| x_5 | 3.6 | 3.6 | 3.6 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 |
| x_6 | 3.6 | 3.6 | 3.6 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 |
| x_7 | ^k NA | 13.22 | 13.22 | ^k NA | 13.20 | 13.20 | ^k NA | 14.90 | 14.91 |
| x_8 | ^k NA | ^k NA | 4 | ^k NA | ^k NA | 5 | ^k NA | ^k NA | 9 |
| C_{TM} | 3863 | 3817 | 3817 | 4617 | 4543 | 4629 | 4754 | 4837 | 5680 |
| P_{LL} | 5012 | 4886 | 4886 | 4265 | 4297 | 4215 | 4092 | 3958 | 3827 |
| P_{NLL} | 856 | 854 | 854 | 685 | 662 | 613 | 737 | 765 | 664 |
| C_{TLT} | 28790 | 28394 | 28394 | 26588 | 26371 | 25882 | 26786 | 26794 | 26844 |
| T_{los} | 5868 | 5740 | 5740 | 4950 | 4959 | 4828 | 4829 | 4723 | 4491 |

^kNA - Not applicable

Table 3. Performance of CMA-ES for TD Optimization Problem

| Objective function | Best value | Worst value | Mean variance | SD | MCT (sec) |
|--------------------|------------|-------------|---------------|------|-----------|
| z_1 | 3817 | 3826 | 3819.8 | 3.82 | 96 |
| z_2 | 25882 | 25882 | 25882 | 0 | 86 |
| z_3 | 4491 | 4491 | 4491 | 0 | 98 |

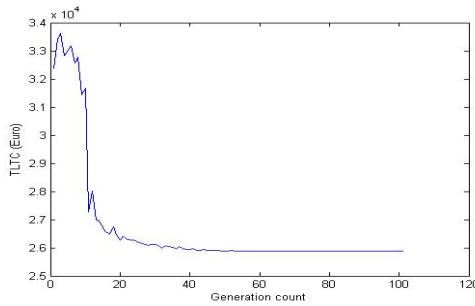


Fig. 1. Convergence plot for z_2 with dv_3

6 Conclusion

In this paper, CMA-ES is employed for the optimum design of three phase distribution transformer. The work proposed aiming at contributing a TD that minimize the objective(s) such as purchase cost, total life-time cost, and total loss of the transformer using proposed MDV, taking into account the constraints imposed by the international standards, transformer specifications and customer needs. The validity of the CMA-ES for solving TD optimization problem is illustrated by its application to a 400KVA distribution transformer design and comparison of its simulation results with CTDTP and BBA-MINLP method in case study 1. Case studies 2 -3 have clearly investigated the significance of MDV for all the TD objective functions and have proven that MDV is efficient for the TD optimization problem. The proposed MDV are not only capable of producing optimum design, but can also render considerable cost savings, and loss reduction. Case study 4 has undoubtedly demonstrated the effectiveness of CMA-ES with respect to its global searching, solution precision, consistency in obtaining solutions, and faster convergence. On comparison with BBA-MINLP method [10] and CDTP [15], CMA-ES dv_3 has given cost savings of about 10.11%, and 17.24% for z_1 , and, 6.12% and 5.088% for z_2 respectively.

Acknowledgements. The authors would like to acknowledge the efforts of Marina A. Tsili, and Eleftherios I. Amoiralis for their valuable assistance in supplying the transformer design specifications.

References

1. Eleftherios, I.A., Marina, A.T., Kladas, A.G.: Transformer design and optimization: A literature survey. *IEEE Transactions on Power Delivery* 24(4), 1999–2024 (2009)
2. Georgilakis, P.S., Doulamis, N.D., Doulamis, A.D., Hatziargyriou, N.D., Kollias, S.D.: A novel iron loss reduction technique for distribution transformers based on a combined genetic algorithm-neural network approach. *IEEE Trans. Syst., Man, Cybern. C* 31, 16–34 (2001)
3. Georgilakis, P.S., Hatziargyriou, N.D., Pappas, D.: AI helps reduce transformer iron losses. *IEEE Comput. AI Power* 12(4), 41–46 (1999)
4. Tsivgouli, A.J., Tsili, M.A., Kladas, A.G., Georgilakis, P.S., Souflaris, A.T., Skarlatini, A.D.: Geometry optimization of electric shielding in power transformers based on finite element method. *Journal of Materials Processing Technology* 181, 159–164 (2007)
5. Subramanian, S., Padma, S.: Optimization of transformer design using bacterial foraging algorithm. *International Journal of Computer Applications* 19(3), 52–57 (2011)
6. Rabih, A.J.: Application of geometric programming to transformer design. *IEEE Transactions on Magnetics* 41(11), 4261–4269 (2005)
7. Padma, S., Bhuvaneswari, R., Subramanian, S.: Optimal design of power transformer using simulated annealing technique. In: *IEEE Conference on Industrial Technology (ICIT)*, Mumbai, India, December 15-17, pp. 1015–1019 (2006)
8. Georgilakis, P.S., Tsili, M.A., Souflaris, A.T.: A heuristic solution to the transformer manufacturing cost optimization problem. *Journal of Materials Processing Technology* 181, 260–266 (2007)

9. Georgilakis, P.S.: Recursive genetic algorithm-finite element method technique for the solution of transformer manufacturing cost minimization problem. *IET Electr. Power AI* 3(6), 514–519 (2009)
10. Amoiralis, E.I., Georgilakis, P.S., Tsili, M.A., Kladas, A.G.: Global transformer optimization method using evolutionary design and numerical field computation. *IEEE Transactions on Magnetics* 45(3), 1720–1723 (2009)
11. Li, H., Han, L., He, B., Yang, S.: Application research based on improved genetic algorithm for optimum design of power transformers
12. Kefalas, T., Tsili, M., Kladas, A.: Unification of anisotropy and FEM-BE models for distribution transformer optimization. *IEEE Transactions on Magnetics* 10(5), 1143–1148 (2008)
13. Amoiralis, E.I., Georgilakis, P.S., Kefalas, T.D., Tsili, M.A., Kladas, A.G.: Artificial intelligence combined with hybrid FEM-BE techniques for global transformer optimization. *IEEE Transactions on Magnetics* 43(4), 1633–1636 (2007)
14. Hansen, N.: The CMA evolution strategy: a comparing review. In: Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E. (eds.) *Towards a New Evolutionary Computation. STUDFUZZ*, vol. 192, pp. 75–102. Springer, Heidelberg (2006)
15. Georgilakis, P.S.: *Spotlight on Modern Transformer Design*. Springer-Verlag London Limited (2009)
16. <http://www.nsc.co.jp/en/product/kind/sheet/product-list.html>
17. <http://www.lri.fr/~hansen/>

Load Information Based Priority Dependant Heuristic for Manpower Scheduling Problem in Remanufacturing

Shantanab Debchoudhury¹, Debabrota Basu¹,
Kai-Zhou Gao², and Ponnuthurai Nagaratnam Suganthan²

¹ Department of Electronics and Telecommunication Engineering,
Jadavpur University, Kolkata-700032

² School of Electrical and Electronic Engineering, Nanyang Technological University,
Singapore- 639798
{sdch10,basudebabrota29}@gmail.com,
{kzgao,epnsugan}@ntu.edu.sg

Abstract. Disassemble scheduling in remanufacturing is an important issue in the current industrial scenario. Allocation of operators for this purpose forms a special class of manpower scheduling problem with an added layer of restrictions. In this paper we propose a set of heuristics that make use of the concept of load information utilization. Several modifications have been incorporated on the basic framework and thorough comparison has been made between the same to devise an efficient way to tackle remanufacturing scheduling problems. The developed method has been put to test on a series of test functions of varying range of difficulty. The results prove the efficiency of these heuristics to solve this scheduling problem.

1 Introduction

One of the basic processes involved in industrial applications, remanufacturing [1] mainly deals with disassembling a complex machine into smaller and simpler subparts. The method helps in identifying root problems which when addressed can help in formation of a new reassembled machine.

From an environmental point of view remanufacturing is considered as the *ultimate form of recycling* and now-a-days it is an interesting topic for researchers [2]. Even remanufacturing is practically used in several countries across the globe for remanufacturing of several products like aerospace, air-conditioning units, bakery equipments, computer and telecommunication equipment, defence equipments, robots, vending machines, motor vehicles and many more. This environmental edge of remanufacturing process [3] has inspired us to deal with this problem.

For any remanufacturing problem, operators are needed to carry out the disassembling. Hence for an efficient allocation of resources an organized scheduling algorithm is of the utmost importance. Manpower scheduling [4-6] is a particular class of such problems when the task at hand is to basically assign a set of human workers with imposed constraints. A broader method would be the classical Job shop scheduling problem [7] when more generalized machines are employed capable of performing at higher efficiencies compared to human labor.

Standard scheduling techniques have been overshadowed by the use of heuristics and meta-heuristics which aim to devise cost effective and efficient means to solve the issue. The latter consist primarily of algorithms like ABC [8], DE [9], GA [10] to employ a population based stochastic process seeking out the best possible combination for an optimal scheduling. Heuristics [11] on the other hand are general dedicated methods that are based on experience and learning.

In this paper our primary aim is to utilize the concept of man power scheduling for disassembly in remanufacturing using heuristics. No such works that we are aware of are in the literature. Hence we have made use of a load information retention technique besides priority assignment. The process of disassembly is depicted through tree architecture. Thus, the problem at hand is to assign operators at each node from an available set of operators with specific processing times for working on that node. The algorithm keeps a track on the load of each operator and accordingly completes the assignment process in decreasing levels of priority.

The paper is organized as follows. Section 2 deals with discussion of the problem structure and objective. In section 3, details of the priority dependant heuristic is put forward. As basic skeleton, a primitive simple greedy heuristic with no load retention is adapted. The advantages of adding the load retention scheme is then discussed and five variations of the same are put up. In section 4, all the frameworks discussed have been put to test in some generated test cases. The experimental background is mentioned in the same section as well as the obtained results with discussions and figures. The conclusion is finally drawn in Section 5.

2 Problem Model and Objective

The process of disassembly as mentioned in Section 1 is shown using tree architecture. The root node is the main machine which is to be disassembled into subparts. At each level of disassembly a certain set of operators are employed from an available pool and each operator has a value of processing time to operate on that particular level.

Fig 1 shows a standard tree and the corresponding tree information table. As is evident in the figure node 1 or the root node is demarcated as NA since it has no parent. Thus Parent number 1 beside node number 2 shows that node number 1 is parent to node 2.

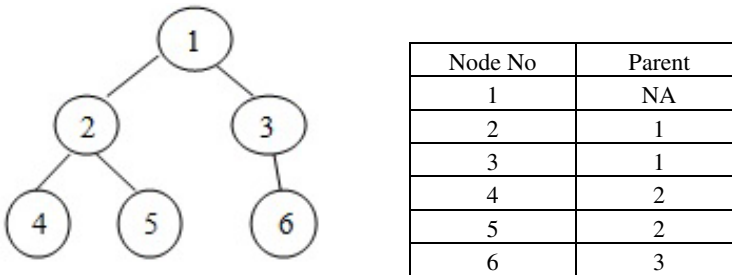


Fig. 1. Sample Tree and corresponding Tree Information Table

We now proceed to show a sample benchmark tree information table in Table 1 that has been utilized as a test case. The problem shown is a 4-operator remanufacturing scheduling problem and has been marked as Instance 1 (Simple) in the subsequent discussions. The presence of '-' indicates that the corresponding operator is not available for operation in the particular shift.

Table 1. Sample Test Case- Instance 1(Simple)

| Node | Parent | Operator 1 | Operator 2 | Operator 3 | Operator 4 |
|------|--------|------------|------------|------------|------------|
| 1 | N.A. | - | - | - | - |
| 2 | 1 | - | - | 15 | 10 |
| 3 | 1 | 5 | - | 10 | - |
| 4 | 2 | 6 | 9 | - | 8 |
| 5 | 2 | - | - | 7 | 10 |
| 6 | 2 | - | 5 | 6 | - |
| 7 | 3 | - | 8 | - | 7 |
| 8 | 3 | 11 | - | 8 | 10 |

Our objective here is to derive a schedule so that makespan [12] or the maximum time for the completion of the entire process is minimized. A schedule which gives the least makespan of all the possible schedules is said to be an optimal schedule. Hence we attempt to allocate in such a way that the schedule obtained is as close to being an optimal schedule as possible.

3 Priority Dependand Heuristic

The remanufacturing model bears significant resemblance to a tree. In discussions henceforth we shall be frequently to a disassembled part to be a child of its former state whom we refer to as the parent. The level of disassembling or in other words the stage up to which the original machine has been disassembled is referred to as the depth. With these annotations in mind, we shall proceed to the assumptions and discussions of the priority dependand heuristic for operator assignment.

The assumptions used for the heuristics designed are as follows:

1. Level with lower amount of disassembled parts is considered for allocation prior to a level with higher amount of disassembled parts. In other words priority of assignment decreases as the depth of the tree increases. We refer to this as a LDHP (Lower Depth Higher Priority) assignment.
2. Children or in other words, disassembled parts of Jobs which are assigned the least time in a certain level are considered first for assignment in the next level. The job whose children are being considered remains unaffected by its own parentage. To put in simple words, process flow of all the heuristics is unidirectional along increasing depth.

3.1 Simple Greedy LDHP Heuristic without Load Information

The simple Greedy Lower Depth Higher Priority (LDHP) algorithm is a primitive attempt to solve remanufacturing scheduling problem. This does not involve any prior

load information of the operators and hence this is not an adaptive algorithm. The greedy based selection of operators is utilized when assignment within a same level of priority is under question. In such a scenario, only that job is taken up which permits allocation of operator with the least possible operating time. Hence it is termed as a greedy selection. The entire process flow of the heuristic is outlined in the Fig 2.

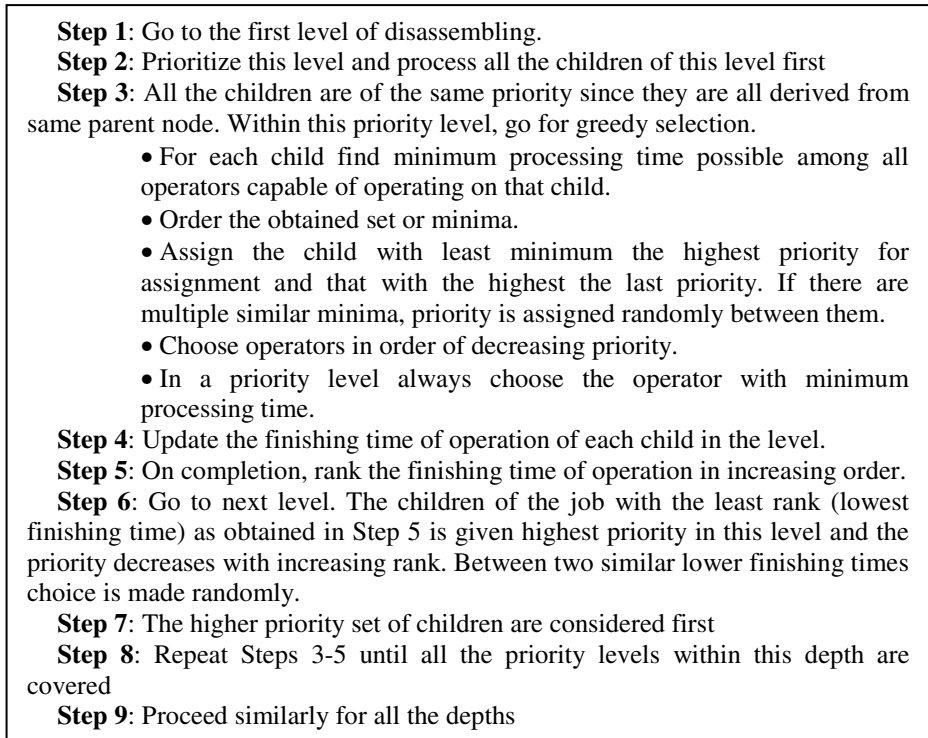


Fig. 2. Process flow for the Simple Heuristic: (Greedy LDHP algorithm)

3.2 Modified Load Information Based LDHP Heuristic with Five Variations

Load Information Retention involves using an adaptive process to determine the existing load on a particular operator. The existing load is added to the offered load in each level of assignment to get a complete picture of the existing scenario, which we refer to here as the Complete Information Schedule (CIS). Real world problems often deal with similar situations when scheduling constraints often demand exclusion of an operator which is already on a certain amount of load. The total time of processing can be taken as a suitable measure of the load and hence this concept is used.

The algorithmic variations from the simple heuristic discussed in Section 3.1 are highlighted in Step 3 of Fig 3. The method of assigning operators in a same priority situation is done using one of five methods discussed subsequently in Fig 4, Fig 5, Fig 6, Fig 7 and Fig 8. The methods or variations deal with choosing between same

priorities and the modes of selecting constitute the basis of the variations. These modes are random, greedy, size, greedy and size, and finally a random pick between the above mentioned four modes.

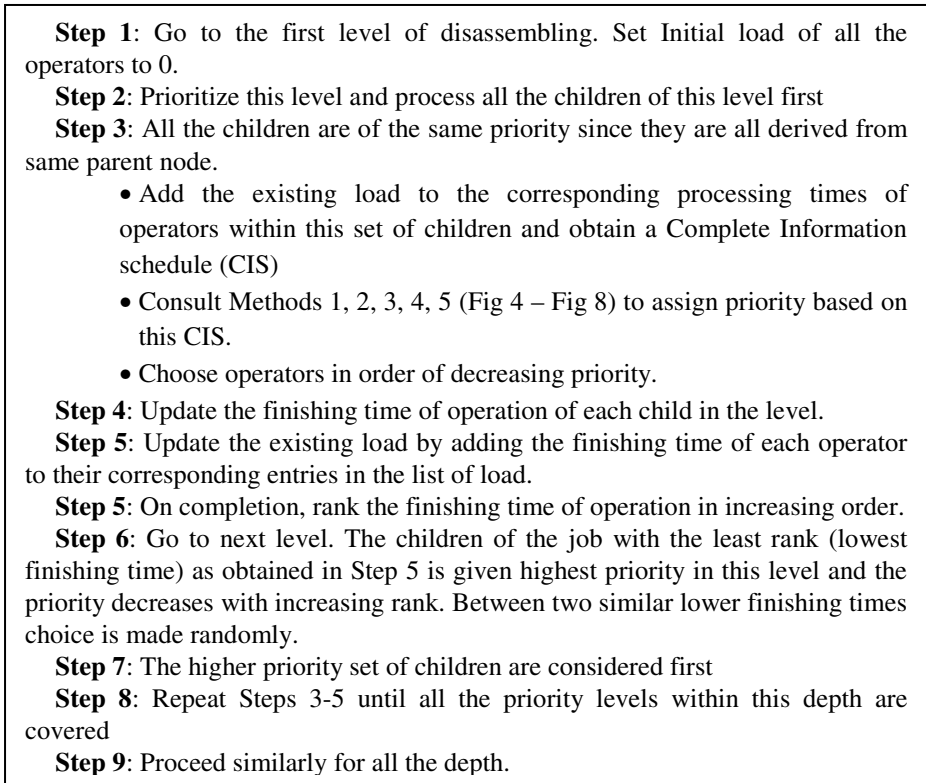


Fig. 3. Process flow for the Load Information based LDHP heuristic

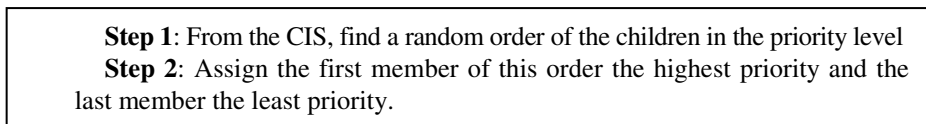


Fig. 4. Method 1: (R-SP) [Random based Priority assignment in Same Priority level]

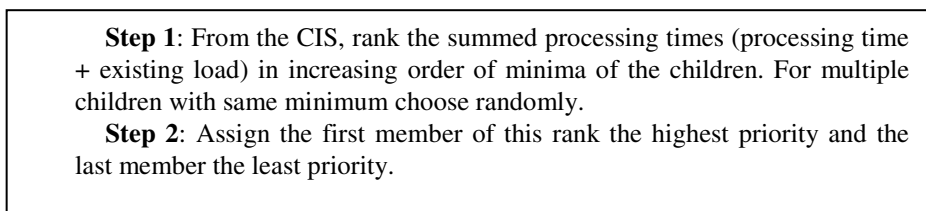


Fig. 5. Method 2: (G-SP) [Greedy based Priority assignment in Same Priority level]

Step 1: From the CIS, rank the size list of operators available of operation in increasing order. For multiple children with same size choose randomly.
Step 2: Assign the first member of this rank the highest priority and the last member the least priority.

Fig. 6. Method 3: (S-SP) [Size based Priority assignment in Same Priority level]

Step 1: From the CIS, list the summed processing times (processing time + existing load) of the children and add the size of the corresponding list of operators available. Rank this combined minimum and size parameter in increasing order. For multiple children with same sum choose randomly.
Step 2: Assign the first member of this rank the highest priority and the last member the least priority.

Fig. 7. Method 4: (GS-SP) [Greedy and Size based Priority assignment in Same Priority level]

Choose any of the R-SP, G-SP, S-SP, GS-SP (Methods 1-4) in random manner

Fig. 8. Method 5: (RCS-SP) [Random chosen scheme based Priority assignment in Same Priority level]

4 Experimental Settings, Results and Discussions

The developed heuristics have been put to test through a series of 10 instances with wide range of difficulty. Four are relatively easy scheduling problems where number of operators is limited to a maximum of 6 and depth of disassembling is at most 4. Four cases require assignment of medium level of difficulty where depth of tree is at most 7 and number of operators involved is at most 10. The remaining two cases pose a difficult assignment challenge with involvement of 15 operators and a depth level of 10. The simple instance 1 has been shown in Table 1 in Section 2. In each of the cases, the makespan is noted besides the computational run time. Our objective is to minimize the makespan.

Table 2 and Table 3 show the results of the simple Greedy LDHP heuristic along with the Load information based LDHP algorithm with all the five methods of assignment within a same priority level. In addition two additional hybrid algorithms are devised.

In “Hybrid Pop based complete heuristic” a population pool of 5 is chosen and each is tested with Load information based assignment with one of the five methods, one after the other. The five different methods work together and the best results are reported. However in “Hybrid Pop based R-G-GS algorithm” only 3 membered populations with R-SP, G-SP and GS-SP modes of selection are used in each of the populations. Again the best results are obtained.

A total of 10 runs have been taken for each instance and each algorithm. The best results have been adopted in each case. All the codes have been run on Intel Core™ i3 machine with 2.26 GHz processor and 3GB memory on MATLAB 2012 platform.

Table 2. Results obtained under Instances 1-4

| HEURISTIC | | Instance 1 (Simple) | Instance 2 (Simple) | Instance 3 (Simple) | Instance 4 (Simple) |
|--|----------------|------------------------|------------------------|------------------------|------------------------|
| Simple Greedy LDHP Heuristic | Makespan | 20 | 45 | 45 | 41 |
| | Run timee(sec) | 5.96e-02 | 6.31e-02 | 7.50e-02 | 8.01e-02 |
| Load Info based Assignment (R-SP) | Makespan | 20 | 35 | 36 | 41 |
| | Run time(sec) | 6.20e-02 | 5.25e-02 | 5.09e-02 | 9.37e-02 |
| Load Info based Assignment (G-SP) | Makespan | 20 | 39 | 29 | 41 |
| | Run time(sec) | 6.01e-02 | 6.00e-02 | 6.03e-02 | 6.87e-02 |
| Load Info based Assignment (S-SP) | Makespan | 20 | 39 | 29 | 41 |
| | Run time(sec) | 4.99e-02 | 5.88e-02 | 5.50e-02 | 6.34e-02 |
| Load Info based Assignment (GS-SP) | Makespan | 20 | 39 | 29 | 41 |
| | Run time(sec) | 5.24e-02 | 6.32e-02 | 5.41e-02 | 6.88e-02 |
| Load Info based Assignment (RCS-SP) | Makespan | 20 | 39 | 33 | 41 |
| | Run time(sec) | 1.18e-01 | 9.09e-02 | 5.99e-02 | 6.97e-02 |
| Hybrid Pop based Complete heuristic | Makespan | 20 | 35 | 29 | 41 |
| | Run time(sec) | 8.87e-01 | 7.98e-01 | 7.86e-01 | 6.63e-01 |
| Hybrid Pop based R-G-GS heuristic | Makespan | 20 | 35 | 29 | 41 |
| | Run time(sec) | 4.12e-01 | 3.99e-01 | 3.32e-01 | 2.19e-01 |

Table 3. Results obtained under Instances 5-10

| HEURISTIC | | Instance 5 (Medium) | Instance 6 (Medium) | Instance 7 (Medium) | Instance 8 (Medium) | Instance 9 (Hard) | Instance 10 (Hard) |
|--|---------------|------------------------|------------------------|------------------------|------------------------|----------------------|-----------------------|
| Simple Greedy LDHP Heuristic | Makespan | 135 | 110 | 116 | 114 | 198 | 189 |
| | Run time(sec) | 9.33e-02 | 1.23e-01 | 1.11e-01 | 1.11e-01 | 1.77e-01 | 1.89e-01 |
| Load Info based Assignment (R-SP) | Makespan | 101 | 83 | 85 | 83 | 128 | 157 |
| | Run time(sec) | 1.17e-01 | 1.23e-01 | 8.27e-02 | 9.66e-02 | 1.58e-01 | 1.97e-01 |
| Load Info based Assignment (G-SP) | Makespan | 100 | 86 | 86 | 87 | 125 | 157 |
| | Run time(sec) | 9.93e-02 | 1.07e-01 | 9.72e-02 | 9.28e-02 | 1.67e-01 | 1.75e-01 |
| Load Info based Assignment (S-SP) | Makespan | 104 | 77 | 91 | 89 | 126 | 158 |
| | Run time(sec) | 3.09e-01 | 1.26e-01 | 9.88e-02 | 9.11e-02 | 1.69e-01 | 1.75e-01 |
| Load Info based Assignment (GS-SP) | Makespan | 101 | 77 | 82 | 82 | 129 | 161 |
| | Run time(sec) | 1.10e-01 | 1.09e-01 | 1.03e-01 | 6.89e-02 | 2.10e-01 | 1.50e-01 |
| Load Info based Assignment(RCS-SP) | Makespan | 105 | 81 | 87 | 86 | 126 | 160 |
| | Run time(sec) | 1.12e-01 | 1.10e-01 | 1.26e-01 | 1.09e-01 | 1.71e-01 | 1.93e-01 |
| Hybrid Pop based Complete heuristic | Makespan | 100 | 77 | 82 | 82 | 125 | 157 |
| | Run time(sec) | 1.12e+00 | 9.88e-01 | 7.97e-01 | 8.55e-01 | 1.03e+00 | 1.63e+00 |
| Hybrid Pop based R-G-GS heuristic | Makespan | 100 | 77 | 82 | 82 | 125 | 157 |
| | Run time(sec) | 5.41e-01 | 4.13e-01 | 3.32e-01 | 3.11e-01 | 6.98e-01 | 7.01e-01 |

The above results show that the simple greedy heuristic is basically a primitive model which fails to address the necessities of a minimal makespan. A marked improvements in the results is obtained when we include the load information retention mechanism. Although all the five variations of this Load information based heuristic are remarkably superior to their primitive skeleton – the greedy simple heuristic, it is however seen that on comparison between themselves slight differences arise. In some instances (Instance 2) R-SP mode works the best while on others G-SP (Instance 9) or in some cases (Instance 7) GS-SP outperforms the other heuristics.

Hence hybrid algorithms were devised. The hybrid pop based complete heuristic manages to achieve minimum makespan on all the cases albeit at the cost of a much increased computational time as compared to the others. The hybrid pop based R-G-GS heuristic is finally seen to be an optimal algorithm since it gives the minimum makespan with a much lesser computational time as compared to the other hybrid algorithm. The Gantt Chart [13, 14] is depicted below in Fig 9 for the best solution obtained in case of Simple Instance 1. The horizontal axis shows time units elapsed while vertical axis shows node number on which operation takes place.

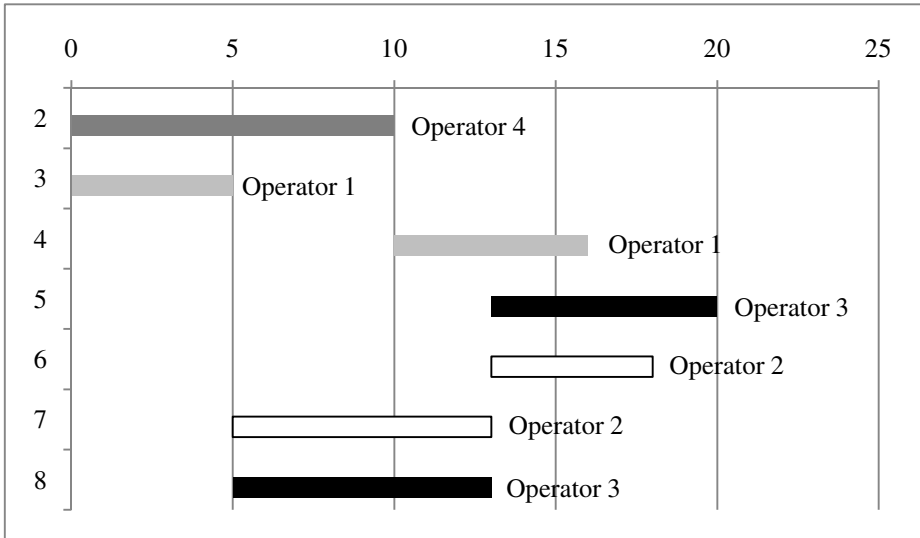


Fig. 9. Gantt chart corresponding to solution of Table 1

5 Conclusions

Before concluding our discussion we quickly recapitulate the major aspects of the heuristic proposed in our paper

First and foremost, the load information scheme is shown to be a fast performing and highly dependent method of obtaining scheduling solutions. The heuristic is derived from a grass root level and it has been shown that incorporation of load information retention scheme improves its performance to a great extent. The modifications or variations involved in the heuristic are all comparable in terms of performance although certain variations work well in some instances. Hybrid variations thus devised were seen to be highly efficient although compromises were made with the algorithmic run time. Summing it all up, such a proposed heuristic assures a solution that boasts of qualities of reliability robustness and effectiveness.

Hence the load information based priority dependant heuristic is found to be a dependable method to solve manpower scheduling based remanufacturing problems.

References

- [1] Remanufacturing, link, <http://en.wikipedia.org/wiki/Remanufacturing>
- [2] The Remanufacturing institute, link, http://reman.org/AboutReman_main.htm
- [3] Remanufacturing, <http://www.remanufacturing.org.uk/>
- [4] Lau, H.C.: On the complexity of manufacturing shift scheduling. *Computers Ops. Res.* 23(1), 93–102 (1996)
- [5] Ho, S.C., Leung, J.M.Y.: Solving a manpower scheduling problem for airline catering using metaheuristics. *European Journal of Operational Research* 202, 903–921 (2010)
- [6] Pan, K.-Q., Suganthan, P.N.: Solving manpower scheduling problem in manufacturing using mixed-integer programming with a two-stage heuristic algorithm. *Int. J. Adv. Manuf. Technol.* 46, 1229–1237 (2010)
- [7] Joseph, A., Egon, B., Daniel, Z.: The Shifting Bottleneck Procedure for Job-shop Scheduling. *Management Science* 34(3) (March 1988) (printed in USA)
- [8] Wang, L., Zhou, G., Xu, Y., Wang, S., Liu, M.: An effective artificial bee colony algorithm for the flexible jobshop scheduling problem. *The International Journal of Advanced Manufacturing Technology* 60(1-4), 303–315 (2012)
- [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] Pezella, F., Morganti, G., Ciaschetti, G.: A genetic algorithm for the flexible Job-shop Scheduling Problem. *Computers & Operations Research* 35(10), 3202–3212 (2008)
- [11] Heuristics, <http://en.wikipedia.org/wiki/Heuristic>
- [12] Chekuri, C., Bender, M.: An efficient Approximation Algorithm for Minimizing Makespan on Uniformly Related Machines. *Journal of Algorithms* 41(2), 212–224 (2001)
- [13] Geraldi, J., Lechter, T.: Gantt charts revisited: A critical analysis of its roots and implications to the management of projects today. *International Journal of Managing Projects in Business* 5(4), 578–594 (2012)
- [14] Gantt chart link, http://en.wikipedia.org/wiki/Gantt_chart

A Tree Based Chemical Reaction Optimization Algorithm for QoS Multicast Routing

Satya Prakash Sahoo, Sumaiya Ahmed, Manoj Kumar Patel, and Manas Ranjan Kabat

Department of Computer Science and Engineering
Veer Surendra Sai University of Technology, Burla, India
{sa_tyaprakash, sumaiyahmed28, patel.mkp}@gmail.com,
manas_kabat@yahoo.com

Abstract. The onset of various real-time multimedia applications in whirlwind networks prompt the necessity of QoS based multicast routing. A multicast communication creates a distribution tree structure, on which a multicast source sends a single copy of data to a group of receivers. QoS multicast routing is a non-linear combinatorial optimization problem which has been proved to be NP-complete, in which we try to find a multicast tree with minimized cost that satisfies multiple constraints. In this paper, we propose a tree based Chemical Reaction Optimization (CRO) Algorithm to solve QoS multicast routing problem. CRO mimics trees as molecules in a chemical reaction, and performs collision between molecules and based on potential energy, we select optimal result. Reckoning results for various random generated networks show that the proposed algorithm outperforms other existing heuristics.

1 Introduction

Multicasting is the ability of communication network to accept a single message from an application and to deliver copies of the message to multiple recipients at different locations [1]. One of the challenges is to minimize the amount of network resources employed by multicasting. Now-a-days many multicast applications exist such as news feeds, file distribution, interactive games and video conferencing. Multimedia applications place new requirements on network as compared to traditional data application They require relatively high bandwidth on a continuous basis for long periods of time and also they involve multipoint communication. Thus these are expected to make heavy use of multicasting, and tend to be interactive. Therefore, it is an important research problem to setup multicast routing quickly and with high QoS. The QoS requirements can be classified into link constraints (e.g. bandwidth), path constraints (e.g. end to end delay, loss rate) and tree constraints (delay jitter). The optimization problem with two or more constraints is known as NP-Complete [2].

Previous researchers have focused on developing heuristic and meta-heuristic algorithms such as PSO [3] [4] [6], GA [17], ACO [8] [16], QPSO [20], Memetic [12-13] etc. These algorithms take polynomial time and produce near optimal results for solving the multicast routing problem. These heuristics can be classified into tree based and path based. In path based heuristics [7] [18], the k -shortest path for each

destination is computed and then the paths to all the destinations are combined to generate the multicast tree. A hybrid PSO with GA operator [7] is used for multicast routing problem. In [7], a set of k -shortest paths are generated as particles and a two point crossover is done on any two randomly chosen particles. It usually happens that this crossover generates a duplicate particle and reduce the searching ability. It again requires an extra effort in removing the duplicate particle and generate a new particle. Sun et al. [18] described an algorithm based on the quantum-behaved PSO (QPSO), for QoS multicast routing. The proposed method converts the QoS multicast routing problem into an integer-programming problem and then solves the problem by QPSO.

In tree based heuristics [6] [8] [17], the tree is directly generated by heuristic algorithms, which reduces the effort of generating k -shortest paths and combining them. A tree based PSO has been proposed in [6] for optimizing the multicast tree directly. However, the performance depends on the number of particles generated. Another drawback of the algorithm is merging the multicast trees, eliminating directed circles and nested directed circles are very complex. A tree growth based ACO algorithm has been proposed in [8] to generate a multicast tree in the way of tree growth and optimizing ant colony parameters through the most efficient combination of various parameters. The general weakness of ant colony algorithm is that it converges slowly at the initial step and takes more time to converge which is due to improper selection of initial feasible parameter. The overhead also increases due to merging and pruning of trees. A Tree Based Swarm Intelligent Algorithm (TBSIA) is proposed in [17] for solving QoS multicast routing problem. The TBSIA presents two collective and co-ordination process for the mobile agents. One is based on the ACO [8] algorithm for guiding the agents' movements by pheromones in the shared environment locally and the PSO algorithm [6] for obtaining the global maximum of the attribute values through the random interaction between the agents.

In this paper, we develop a Tree Based Chemical Reaction Optimization algorithm (TBCROA) for solving the QoS multicast routing problem. The TBCROA generated a number of multicast trees. Each tree represents a molecule and then the inter-molecular ineffective collision and on-wall ineffective collision to generate new molecules. The algorithm terminates finally generating the best molecule.

The rest of the paper is organized as follows. The problem description and formulation is given in section 2. The basic concept of Chemical Reaction Optimization (CRO) and TBCROA is presented in section 3. The section 4 gives the performance evaluation of the proposed algorithm with respect to the other existing algorithms. Finally, the concluding remarks are presented in Section 5.

2 Problem Statements and Formulation

The problem of multicast routing in communication network is equivalent to finding a multicast tree T in graph G such that T spans the source and all members of in the multicast group M . A network is represented as an undirected graph $G(V, E)$, where V is the set of vertices i.e. nodes and E is the set of edges i.e. communication link between the nodes. Each link $(u, v) \in E$ in G has four weights $bw(u, v)$, $delay(u, v)$,

$cost(u, v)$ and $loss_rate(u, v)$ which represent available bandwidth, the delay, the cost and the loss probability. Let $s \in V$ be the source node and $M \subseteq V - \{s\}$ is the set of multicast destinations.

The cost of a multicast tree is the sum of the costs of all the links in the multicast tree. A good multicast tree tries to minimize this cost. The total cost of the tree $T(s, M)$ can be given by

$$C(T(s, M)) = \sum_{(u,v) \in T(s,M)} cost(u, v) \quad (1)$$

The end to end delay from the source node to the destination node is the sum of the individual link delays along the route. A good multicast tree tries to minimize the end to end delay for every source-destination pair in the multicast group. The total delay of the path $P_T(s, m)$ where $m \in M$ can be defined as

$$D(P_T(s, m)) = \sum_{(u,v) \in P_T(s,m)} delay(u, v) \quad (2)$$

The total delay of the tree $T(s, M)$ is defined as maximum of the delay on the paths from source to each destination.

$$D(T(s, M)) = \max_{m \in M} \{D(P_T(s, m))\} \quad (3)$$

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

$$B(P_T(s, m)) = \min\{bw(u, v) | (u, v) \in P_T(s, m)\} \quad (4)$$

The total loss probability of the path can be defined as

$$L(P_T(s, m)) = 1 - \prod_{(u,v) \in P_T(s,m)} (1 - (loss_rate(u, v))) \quad (5)$$

The delay jitter is defined as the average difference of the delay on the paths from source to destination node.

$$D_j(T(s, M)) = \sqrt{\sum_{m \in M} (D(P_T(s, m)) - (delay_{avg}))^2} \quad (6)$$

Where $delay_{avg}$ is the average delay calculated from source to all destinations.

The multi-constrained least cost multicast problem is defined as:

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

$$\left\{ \begin{array}{l} D(P_T(s, m)) \leq \delta \text{ end-to-end delay requirement} \\ L(P_T(s, m)) \leq \mathcal{L} \text{ loss probability} \\ B(P_T(s, m)) \geq \varepsilon \text{ minimum bandwidth} \\ D_j(T(s, M)) \leq \vartheta \text{ jitter delay constraint} \end{array} \right. \quad (7)$$

A good multicast tree is scalable in two respects i.e. constructing a multicast tree for a large multicast group should require reasonable amounts of time and resources. The second is that the routers in the communication network should be able to simultaneously support a large number of multicast trees.

3 A Tree Based CRO for QoS Multicast Routing Problem

In this paper, we propose a tree based CRO algorithm (TBCROA) for QoS multicast routing problem. The TBCROA generates a number of multicast trees by a heuristic algorithm. Each tree is considered as a molecule and fitness of the multicast tree is considered as the potential energy of the molecule. Then we use Inter-Molecular ineffective collision to generate new molecules. The less stable molecules are removed after each iteration. This process continues for a fixed number of iterations to come up with the most stable molecule. The basic concept of CRO, representation of molecule, calculation of potential energy of molecule and operation of the TBCROA is presented in the following subsections.

3.1 Chemical Reaction Optimization (CRO)

The CRO [9-11] [14] is a recently established meta-heuristic for optimization, inspired by the nature of chemical reactions. In a chemical reaction, the initial species/ reactants/ molecules in the unstable states undergo a sequence of collisions and become the final products in stable states. It enjoys the advantages of both Simulated Annealing (SA) and Genetic Algorithm (GA). The basic unit in CRO is a molecule which is considered as a single possible solution in the population. Each molecule has potential energy (PE) and the kinetic energy (KE) which refers to the fitness value that characterize the molecule. A chemical change of a molecule is triggered by a collision which may be uni-molecular collision (molecule hits on some external substances) or inter-molecular collision (molecule collides with other molecules). The corresponding reaction change is termed as Elementary Reaction which can be of four kinds: on-wall ineffective collision, decomposition, inter-molecular ineffective collision, and synthesis. In On-wall ineffective collision, a molecule hits the wall and then bounces back which results subtle change in the molecule. If w is original molecule then w' is the resultant molecule after On-wall ineffective collision. In Decomposition, a molecule hits the wall and then decomposes into two or more molecules. If w is original molecule then w_1 and w_2 are resultant molecules. In Inter-molecular ineffective collision, more than one molecule collide with each other and then bounces back which results in two molecules with subtle change. If w_1 and w_2 are existing molecules then w_1' and w_2' are resultant molecules. In Synthesis, more than one molecule collides with each other and combines with each other to form one. If w_1 and w_2 are existing molecules then w' is the resultant molecule. Synthesis is vigorous.

3.2 Representation of the Molecule

We use heuristic approach to construct a set of multicast trees randomly. Each multicast tree $T(V_T, E_T)$ represents a molecule. Where V_T is set of nodes and E_T is the set of edges in T . The multicast tree $T(V_T, E_T)$ is constructed by initializing by $V_T = \{s\}$ and $E = \phi$. The delay up to node $s(dsf(s))$, cost up to node $s(csfs(s))$ and loss up to

node $s(lsf(s))$ are set 0, 0 and 0 respectively. The step by step procedure for the construction of multicast tree is as follows:

Step 1: The source s is considered as the current node.

Step 2: For each unvisited neighboring node 'v' of current node the delay, bandwidth and loss rate constraints are checked i.e.

$$\begin{cases} B(curr_node, v) > \varepsilon \\ dsf(curr_node) + D(curr_node, v) < \vartheta \\ 1-(1-lsf(curr_node))(1-L(curr_node, v)) < \xi. \end{cases} \quad (8)$$

Step 3: The neighboring nodes that satisfy these conditions are stored in an array along with their predecessors and the cost up to those nodes csf are taken as their priorities.

Step 4: The node j with lowest csf value is taken as the current node and the $csf(j)$, $dsf(j)$, $lsf(j)$ are calculated based on the equations between predecessor node to the considered neighboring node.

Step 5: If the considered current node is our multicast destination, then the $csf(j)$, $dsf(j)$, $lsf(j)$ are initialized to zero and from current node to neighboring node update the cost is updated.

Step 6: Then the step 1 to step 5 are repeated for the updated current node until V_T contain all nodes of the multicast group.

Step 7: After tree is constructed we check whether tree satisfies the delay jitter constraint i.e. $D_j(T(s, M)) \leq \vartheta$. This is considered as the best tree and the fitness value of the tree is calculated.

Step 8: After the first tree is generated, 10% of the total number of edges of the tree is deleted from the network randomly and from the resulting network graph we will generate next tree using step 1 to step 8.

3.3 Potential Energy of the Molecule

The fitness of the multicast tree is considered as the potential energy of the molecule. The potential energy i.e. fitness function for this algorithm is defined as follows:

$$PE(T(s, M)) = cost(T(s, M)) + \eta_1 \min\{\vartheta - \text{delay}(T(s, M)), 0\} + \eta_2 \min\{\vartheta - \text{jitter}(T(s, M)), 0\} + \eta_3 \min\{\xi - \text{loss}(T(s, M)), 0\} \quad (9)$$

where, η_1, η_2, η_3 are punishment coefficients that decide punishment range. If $b \geq 0$ $\min(b, 0) = b$; else $\min(b, 0) = 0$.

3.4 Operation of the TBCROA

The step by step operation of the TBCROA is presented below.

Step 1: A set of multicast trees are randomly generated by using the procedure presented in section 3.2. Then two trees are selected randomly from this set of trees.

Step 2: The TBCROA collides these two trees using inter- molecular ineffective collision i.e. the edges are swapped between selected nodes and calculate the fitness of newly generated tree.

Step 3: All the multicast trees generated are sorted with respect to their fitness and the less fit trees are destructed.

This process is repeated for a predefined number of iterations or the stopping criteria is met. If the fitness of the global best molecule is not changed after three iterations then that is considered as the best multicast tree.

Pseudo code TREE_BASED_CRO

```

1. Begin
2. Generate the set of multicast trees by using procedure described
   in Section 3.2
3. For
4.   each tree calculate potential energy
5.      $PE(T(s,M)) = \text{cost}(T(s,M)) + \eta_1 \min\{\partial - \text{delay}(T(s,M)), 0\} +$ 
        $\eta_2 \min\{\partial - \text{jitter}(T(s,M)), 0\} + \eta_3 \min\{\mathcal{L} - \text{loss}(T(s,M)), 0\}$ 
6. End For
7. While stopping criteria not satisfied do
8.   For i=1 to no of molecules
9.     Randomly select two molecules  $w_1$  and  $w_2$ 
10.     $[w_1', w_2'] = \text{Intermole\_in\_col}(w_1, w_2)$ 
11.    Generate all the molecules and sort them with respect to
       their fitness.
12.   End For
13.   Remove the redundant and less fit molecules
14. End while
15. The optimal molecule will be stored in best_mole.
```

Pseudo code inter_ineff_coll (M1, M2)

```

1. Input: Molecules  $M1, M2$  with their central energy.
2. Obtain an atom from  $M1, M2$ . Swap all the bonds on that atom
   between both molecules.
3. Calculate  $PE_{w1}$  and  $PE_{w2}$ 
4. Return  $w1, w2$ .
```

4 Performance Evaluation

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

The positions of the nodes are fixed randomly in a rectangle of size 4000 km x 2400 km. The Euclidean metric is then used to determine the distance between each pair of nodes. The network topology used in our simulation was generated randomly using Waxman's topology [16]. Edges are introduced between the pairs of nodes u, v with a probability that depends on the distance between them. The edge probability is given by $P(u, v) = \beta \exp(-l(u, v) / \alpha L)$, where $l(u, v)$ is the Euler distance from node u to v and L is the maximum distance between any two points in the network. The delay, loss rate, band width and cost of the links are set randomly from 1 to 30, 0.0001 to 0.01, 2 to 10 Mbps and 1 to 100 respectively.

The source node is selected randomly and destination nodes are picked up uniformly from the set of nodes chosen in the network topology. The delay bound, the delay jitter bound and the loss bound are set 120ms, 60ms and 0.05 respectively. The bandwidth requested by a multicast application is generated randomly. We also implement PSOTREE [6] and TGBACA [8] and TBSIA [17] algorithms in the same environment to study and compare the performance of our proposed algorithm with the existing algorithms. We generate 30 multicast trees randomly to study the performance of our algorithm in comparison to the existing algorithms. The simulation is run for 100 times for each case and the average of the multicast tree cost is taken as the output.

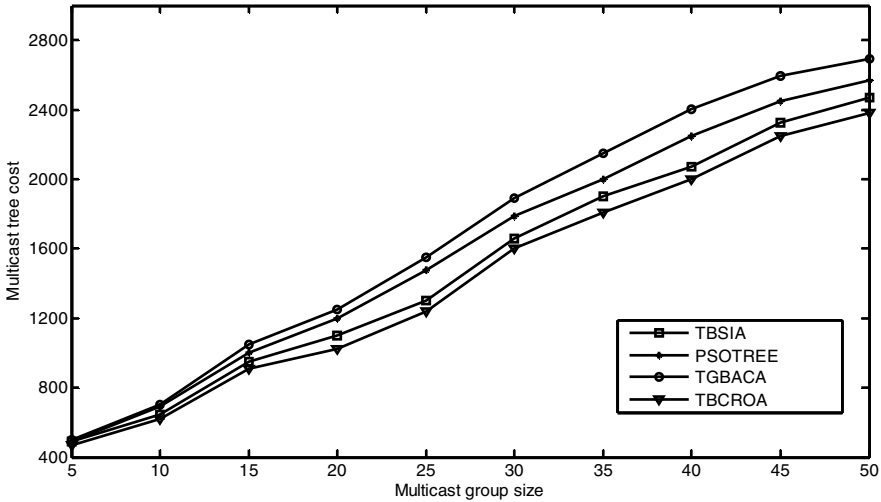


Fig. 1. Multicast tree cost vs. group size (No of nodes=100)

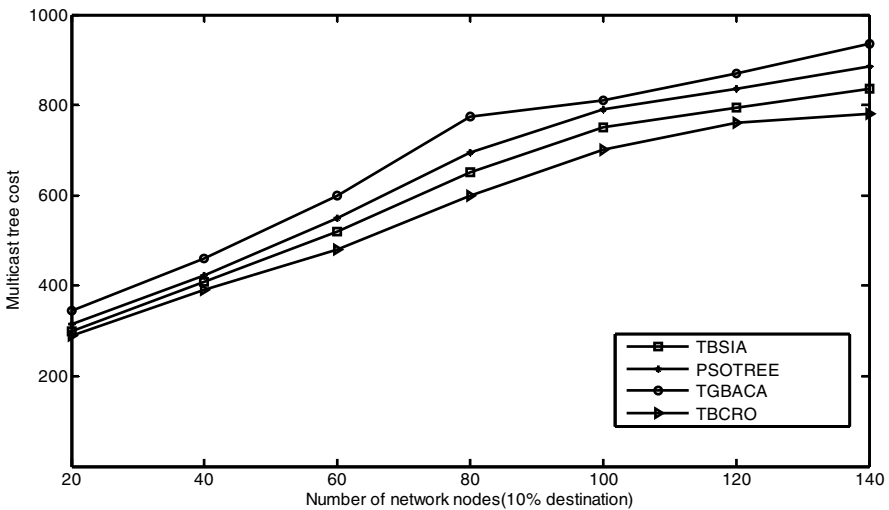


Fig. 2. Multicast tree cost vs. Network size with 10% nodes as destinations

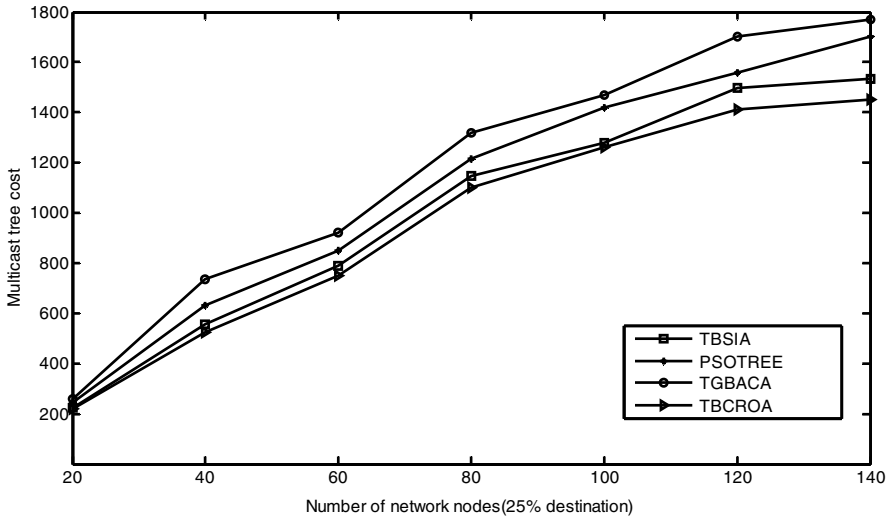


Fig. 3. Multicast tree cost vs. Network size with 25% nodes as destinations

The multicast tree cost versus multicast group size for a network of 100 nodes is shown in Fig. 1. The Fig. 2 shows the multicast tree cost versus the network size with 10 percent of the nodes as the group size. The multicast trees generated by PSOTREE, TGBACA, TBSIA and our proposed algorithm satisfy the delay, delay jitter, loss rate and bandwidth constraints. However, the figures clearly illustrate that the cost of the multicast tree generated by our proposed algorithm is less than the multicast trees generated by PSOTREE and TGBACA and TBSIA. The PSOTREE algorithm constructs the multicast tree by combining the multicast trees and removing directed cycles. This algorithm removes the links that are in any of the trees, but not in both and have minimum fitness. However, this approach may not generate a better tree, because the links deleted from the cycle may be better than the links not in the directed cycles. The TGBACA algorithm follows a pheromone updating strategy to construct the best multicast tree. The algorithm updates pheromones on the links used by the global best tree and the best tree generated after each generation. Though this strategy fasts the convergence process, but the solution may fall into local optimization. The TBSIA combines two multicast tree patterns by bringing the better attributes of one pattern to another pattern. It generates a new tree pattern after each iteration, which is better than both the patterns. Since the whole path of one tree is replaced by another path from another multicast tree, some better links may be excluded from the tree. This may fail to generate an optimal tree in some cases.

The TBCRO algorithm randomly collides two multicast trees which always try to choose the optimal links into the multicast tree. Therefore TBCROA generates the optimal multicast tree in most of the cases.

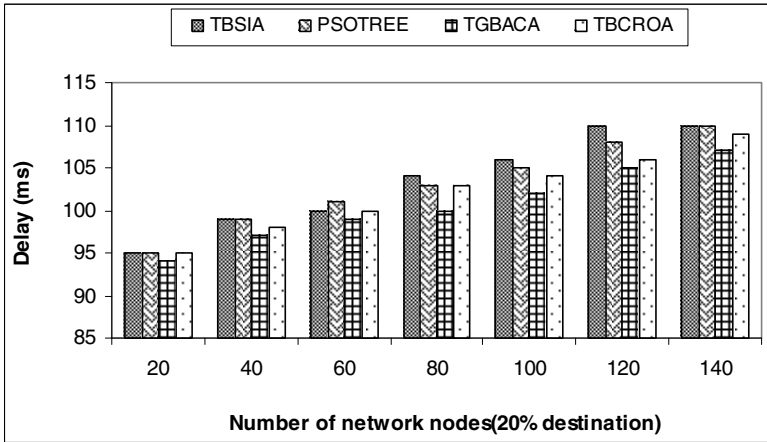


Fig. 4. Multicast tree Delay vs. Network size with 20% nodes as destinations

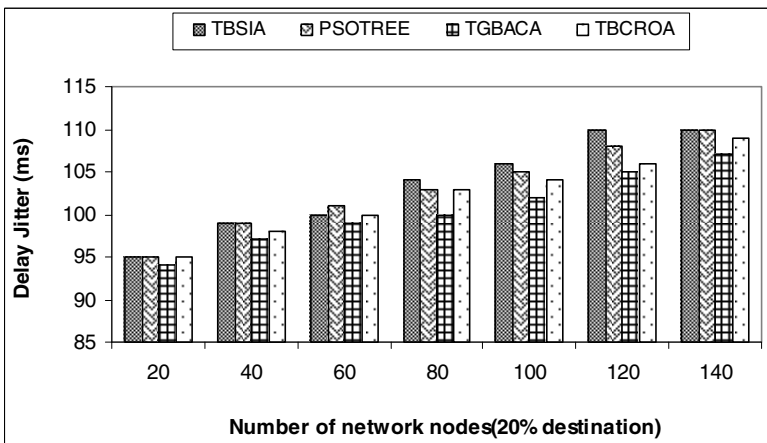


Fig. 5. Multicast tree Delay Jitter vs. Network size with 20% nodes as destinations

The Fig. 4 and Fig. 5 show the multicast tree delay and delay jitter versus number of network nodes with 20 % nodes as destinations respectively. It is observed that the proposed algorithm along with PSOTREE [6] and TGBACA [8] and TBSIA [17] satisfy the delay and the delay jitter constraints. The multicast tree generated by our algorithm experiences a delay and delay jitter comparable with PSOTREE [6] and TGBACA [8] and TBSIA [17]. However, the multicast tree generated by our algorithm performs significantly better in terms of multicast tree cost.

5 Conclusion

This paper presents a tree based chemical reaction optimization algorithm for QoS multicast routing problem. The proposed algorithm generates an economic multicast tree that satisfies delay, delay jitter, bandwidth and loss rate constraints. Our algorithm

constructs the multicast tree by randomly colliding multicast trees. Since the collision is done at random points, the TBCRO algorithm selects better link during random collision in comparison to the existing algorithms. Therefore, our algorithm performs better than the existing algorithms.

References

1. Sahasrabudde, L.H., Mukherjee, B.: Multicast Routing Algorithms and Protocols: a tutorial. *IEEE Network.*, 90–102 (2000)
2. Striegel, A., Mannimaran, G.: A Survey of QoS Multicasting Issues. *IEEE Communications Magazine*, 82–87 (2002)
3. Liu, J., Sun, J., Xu, W.-B.: QoS Multicast Routing Based on Particle Swarm Optimization. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) *IDEAL 2006*. LNCS, vol. 4224, pp. 936–943. Springer, Heidelberg (2006)
4. Wang, Z., Sun, X., Zhang, D.: A PSO-Based Multicast Routing Algorithm. In: *Proceedings of Third International Conference on Natural Computation (ICNC)*, pp. 664–667 (2007)
5. Xi-Hong, C., Shao-Wei, L., Jiao, G., Qiang, L.: Study on QoS Multicast Routing Based on ACO-PSO Algorithm. In: *Proceedings of 2010 International Conference on Intelligent Computation Technology and Automation*, pp. 534–537 (2010)
6. Wang, H., Meng, X., Li, S., Xu, H.: A Tree-Based Particle Swarm Optimization for Multicast Routing. *Computer Network* 54, 2775–2786 (2010)
7. Abdel-kader, R.: Hybrid Discrete PSO with GA Operators for Efficient QoS Multicast Routing. *Ain Shams Engineering Journal* 2, 21–31 (2011)
8. Wang, H., Xu, H., Yi, S., Shi, Z.: A Tree Growth Based Ant Colony Algorithm for QoS Multicast Routing Problem. *Expert Systems with Applications* 38, 11787–11795 (2011)
9. Alatas, B.: A Novel Chemistry Based Metaheuristic Optimization Method for Mining of Classification Rules. *Expert Systems with Applications* 39, 11080–11088 (2012)
10. Lam, Y.S.A., Li, O.K.V.: Chemical Reaction Inspired Meta Heuristic for Optimization. *IEEE Transactions on Evolutionary Computation* 14(3), 381–399 (2010)
11. Lam, Y.S.A., Li, O.K.V.: Chemical Reaction Optimization- a tutorial. *Memetic Computing* 4(1), 3–17 (2012)
12. Donoso, Y., Perez, A., Ardila, C., Fabregat, R.: Optimizing Multiple Objectives on Multicast Networks using Memetic Algorithms. *GESTS International Trans. Computer Science and Engineering* 20(1), 192–204 (2005)
13. Zhang, Q., Wang, Z., Zhang, D.: QoS Multicast Routing Optimization based on Memetic Algorithm. In: *International Conference on Management of e-Commerce and e-Government*, pp. 441–444. IEEE (2008)
14. Lam, A.Y.S., Li, V.O.K.: Real-Coded Chemical Reaction Optimization. *IEEE Transactions on Evolutionary Computation* 16(3), 339–353 (2012)
15. Chu, C.H., Gu, J.H., Hou, X.D., Gu, Q.: A Heuristic Ant Algorithm for Solving QoS Multicast Routing Algorithm. *IEEE, Evolutionary Computation* 2, 1630–1635 (2002)
16. Kumar, D.R., Najjar, W.A., Srimani, P.K.: A New Adaptive Hardware Tree Based Multicast Routing in k-ary n-cubes. *IEEE Transactions on Computers* 50(7) (2001)
17. Patel, M.K., Kabat, M.R., Tripathy, C.R.: A Swarm Intelligence Based Algorithm for QoS Multicast Routing Problem. In: Panigrahi, B.K., Suganthan, P.N., Das, S., Satapathy, S.C. (eds.) *SEMCCO 2011, Part II*. LNCS, vol. 7077, pp. 38–45. Springer, Heidelberg (2011)
18. Sun, J., Fang, W., Xie, Z., Xu, W.: QoS Multicast Routing using a Quantum-Behaved Particle Swarm Optimization Algorithm. *Engineering Applications of Artificial Intelligence*, 123–131 (2010)
19. Randaccio, L.S., Atzori, L.: A Genetic Algorithms Based Approach for Group Multicast Routing. *Journal of Networks* 1(4) (2006)

A New Improved Knowledge Based Cultural Algorithm for Reactive Power Planning

Bidishna Bhattacharya^{1,*}, Kamal K. Mandal², and Niladri Chakraborty²

¹ Techno India, Saltlake, Electrical Engineering Dept., India

² Jadavpur University, Department of Power Engineering, India

bidishna_inf@yahoo.co.in

Abstract. This paper proposes a novel hybrid method for the planning of reactive power problem (RPP). The objective of this paper is to determine the optimum investment required to satisfy suitable reactive power constraints for an acceptable performance level of a power system. Due to the discrete nature of reactive compensation devices, the given objective function leads to a nonlinear problem with combined (distinct and constant) variables. It is solved by a hybrid procedure, aiming to develop the best search features of an iterative algorithm. The performance of the proposed procedure is shown by presenting the numerical results obtained from its application to the IEEE 30-bus test network. The results obtained are compared with evolutionary programming (EP) and Broyden method to determine the efficacy of the proposed method.

1 Introduction

Variable load may have severe effects on the performance of a power system due to inadequate reactive power support and active power transmission capability. Reactive power plays an important role in supporting the real power transfer by maintaining voltage stability and system reliability. Using of compensation devices is the efficient method for reactive power dispatch to keep the voltage level within the acceptable levels throughout the system reducing overall real power loss. A planning of reactive power (RPP) is required to determine the optimal placement of compensating devices minimizing the cost associated with it. These costs can be of one fixed and one variable type. The fixed part is due to the difference between the installation cost with and without a reactive device. On the other hand, the variable cost is mainly due to the active losses in the generator caused by the reactive power. Ideally, the operation should minimize the cost due to the losses in the network and in the generation units, while keeping the system safe. This makes the reactive power problem a non-linear optimization problem with several constraints.

Several methods have applied to the reactive power planning problems over the years. Sachdeva et.al proposed nonlinear programming method for optimal reactive power planning in 1988 [1]. In this context successive quadratic programming is also

* Corresponding author.

presented by Quintana et.al [2]. Through the use of mixed integer programming [3] and decomposition method [4] reactive power was optimized to reduce the network losses. In the year of 1997 DaCosta applied primal-dual method for optimal reactive power optimization [5]. Although these classical methods have proved their flourishing results; they experienced with a limited capability to solve non-linear and non-convex power system problems. Since the RPP problems have a lot of constraints both equality and inequality variables, the meta-heuristic methods are preferred as the solution to these problems give more satisfied results.

In the recent years, several meta-heuristic algorithms have been applied for solving modern power system problems in order to overcome some of the drawbacks of conventional techniques. In the era of 90's Genetic Algorithm (GA) [6], Evolutionary Programming (EP) [7] proved their effectiveness in reactive power optimization problems. Reactive power operational and investment-planning problem is solved by Lee et al. [8] by a Simple Genetic Algorithm (SGA) combined with the successive linear programming method. Liu et.al presented Simulated Annealing and Tabu Search based hybrid method to solve the RPP problem as discussed [9]. Several other algorithm presented in recent years for reactive power planning with various objective functions like Particle Swarm Optimization [10],[11], Differential Evolution (DE) [12] etc. Recently Bacteria Foraging Optimization (BFO) [13] and Ant Colony Optimization (ACO) [14] successfully solved RPP problem. A very new method namely Self-Evolving Brain-Storming Inclusive Teaching-Learning-Based Algorithm is used by KR Krishnanand et.al [15] to reduce fuel cost by reactive power optimization. Also in current year Carlos Henggeler Antunes et.al proposes Greedy Randomized Adaptive Search Procedure (GRASP) for reactive power planning [16].

This paper is concerned with application of Cultural Algorithm (CA) for optimal reactive power dispatch with the cost of reactive power sources consisting of installation and operating costs. Cultural Algorithm (CA) was proposed by Reynolds as a model of social evolution [17], [18]. Having two different evolutionary spaces the cultural algorithm leads to a global optimization technique. Cultural Algorithms have been successfully implemented to various global constrained optimization problems of power system field such as substation planning [19], hydrothermal scheduling [20], Economic load dispatch [21], CEED problem [22] etc.

In this study, a new approach based on Cultural Algorithm (CA) [23] for solving RPP in order to minimize objective function including cost of active and reactive powers produced by generators is presented. The mentioned method is studied on 30-bus IEEE standard network and the results are compared to other meta-heuristic approaches [24] to check whether these results are reasonable and feasible.

2 Problem Formulation

Among one of the most important issues of power system, system loss is considered one of the main subject. Through reactive power compensation these system losses can be minimized to optimize the steady performance of a power system in terms of one or more objective functions satisfying several constraints. Being a complex

problem, RPP requires the simultaneous minimization of two objective functions which can be described as follows.

2.1 Minimization of Operation Cost

The first objective deals with the minimization of operation cost by reducing real power loss and improving the voltage profile. So the objective function represents the total cost of energy loss as,

$$Wc = h \sum_{l \in N_l} d_l P_{loss,l} \quad (1)$$

where h is the per unit energy cost, N_l is the load level duration and $P_{loss,l}$ is the real power loss of the network during the period of load level. The $P_{loss,l}$ can be expressed for the duration d_l in the following equation as,

$$P_{loss} = \sum_{k=1}^{NL} G_k (V_i^2 + V_j^2 - 2V_i V_j \cos(\theta_i - \theta_j)) \quad (2)$$

where, G_k is the line conductance of the k th line and V_i and V_j are the voltage magnitude of i th bus and j th bus respectively. Again θ_i and θ_j are the phase angles of i th bus and j th bus respectively.

2.2 Minimization of Allocation Cost

The other objective is to minimize the allocation cost by optimizing the additional reactive power sources. So the second objective function represents the cost of VAR source installation which comprises two components namely fixed installation cost and purchase cost. It can be expressed as,

$$I_C = \sum_{i \in NC} (e_i + C_{ci} QC_i) \quad (3)$$

Where e_i is the pre-specified fixed VAR source installation cost at bus i , C_{ci} is the per unit VAR source purchase cost at bus i and QC_i is the VAR source installation at bus i . NC is the number of VAR source installed buses.

So, the desired objective model can be formulated as,

$$\text{Minimize } fc = I_C + W_C \quad (4)$$

Subject to equal constraint,

$$0 = Q_i - V_i \sum_{j \in N_i} V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \quad i \in N_{PQ} \quad (5)$$

and inequality constraints as follows;

- i. Generator constraints

All Generators voltages and reactive power outputs must be restricted within their permissible lower and upper limits as follows:

$$V_{G_i}^{\min} \leq V_{G_i} \leq V_{G_i}^{\max}, i = 1, 2, \dots, NG \quad (6)$$

$$Q_{G_i}^{\min} \leq Q_{G_i} \leq Q_{G_i}^{\max}, i = 1, 2, \dots, NG \quad (7)$$

where NG is the number of generators.

ii. Load voltage constraints

Load voltages should follow the restriction below,

$$V_{L_i}^{\min} \leq V_{L_i} \leq V_{L_i}^{\max}, i = 1, 2, \dots, NPQ \quad (8)$$

iii. Reactive power generation constraints of capacitor banks

Capacitor bank generation must be within the permissible values as,

$$QC_i^{\min} \leq QC_i \leq QC_i^{\max}, i = 1, 2, \dots, NC \quad (9)$$

iv. Transformer tap setting constraints

Transformer On-load tap changers are bounded as per the settings like,

$$T_i^{\min} \leq T_i \leq T_i^{\max}, i = 1, 2, \dots, NT \quad (10)$$

The equality constraints are satisfied by running the load flow program and the control variables are self controlled by the optimization algorithm. The security constraints like load bus voltages and line flow limit have to be derived as dependent variables. Normally, those candidate solutions which result in violation of upper or lower limits of the above stated dependent variables have to be penalized to discard the infeasible solutions. So, these state variables are satisfied by adding a penalty terms in the objective function. Embedding of the penalty function, the objective function (4) may be modified to,

$$Minimize F_C = fc + \gamma_v \sum_{i=1}^{NPQ} (V_{L_i} - V_{L_i}^{\lim})^2 + \gamma_q \sum_{i=1}^{NG} (Q_{gi} - Q_{gi}^{\lim})^2 \quad (11)$$

where, γ_v, γ_q are the penalty factors. $V_{L_i}^{\lim}$ and Q_{gi}^{\lim} are calculated as,

$$V_{L_i}^{\lim} = \begin{cases} V_{L_i}^{\max}, & V_{L_i} > V_{L_i}^{\max}; \\ V_{L_i}^{\min}, & V_{L_i} < V_{L_i}^{\min}; \\ 0, & \text{else}; \end{cases} \quad (12)$$

$$Q_{gi}^{\lim} = \begin{cases} Q_{gi}^{\max}, & Q_{gi} > Q_{gi}^{\max}; \\ Q_{gi}^{\min}, & Q_{gi} < Q_{gi}^{\min}; \\ 0, & \text{else}; \end{cases} \quad (13)$$

3 Cultural Algorithm (CA)

In 1994 Reynolds first proposed Cultural Algorithm (CA) as a social evolutionary process [18]. This high level searching technique is applied nowadays in many optimization problems [19], [20], [21], [22]. The basics behind CA can be understood from elsewhere [18].

Population space of CA is the space where the information about individuals is stored and influences the belief space where the culture knowledge as a set of promising variable ranges are formed and maintained during the evolution of the population. After evolution of population space with a performance function the acceptance function will determine which individuals are kept aside for Belief space. Experiences of those elite individuals will update the knowledge of the Belief space via update function. This updated knowledge is used to influence the evolution of the population.

A pseudo- code description of cultural algorithm is described as follows,

```

Begin
    t = 0
    Initialize Pt
    Initialize Bt
    Repeat
        Evaluate Pt
        Update (Bt, accept (Pt))
        Generate (Pt, influence (Bt))
        t = t + 1
        Select Pt from Pt-1
    Until (Termination condition achieved)
End
End

```

4 Hybrid Cultural Algorithm

After the application of the basic cultural operators, a problem-specific operator is applied to produce the new generation. In this improved hybrid cultural algorithm, crossover is the main evolutionary operator for the exploitation of information [23]. This approach is capable of locating the global optimal solution promptly. The basic idea of using CA with evolutionary programming with crossover is to influence the mutation operator so that the current knowledge stored in the search space can be properly exploited. Crossover is a mixing operator that combines genetic material from selected parents to improve the search process. Cultural Algorithm belongs to the class of evolutionary algorithms which offers a unique strategy for optimization which is described as follows.

4.1 Crossover

The crossover operator is mainly responsible for the global search. The operator basically combines substructures of two parents to produce new structures. In this work

single point crossovers is employed in which one crossover site is chosen and offspring are created by swapping the information between the chosen crossover sites. This operator acts like a refined crossover operator which searches for improved performance, exploits new areas of the search space far away from the current solution, and retains the diversity of the population.

The value of crossover constant (CR) must be made in the range of [0, 1]. A crossover constant of one leads the trial vector composed with entire mutant vector parameters. However, a crossover constant near zero gives more probability of having parameters from the target vector in the trial vector which can be stated as

$$x''_{i,j} = \begin{cases} x'_{i,j} & \text{if } rand_j \leq CR \\ x_{i,j} & \text{otherwise} \end{cases}$$

Where *rand* is a uniformly distributed random number having the value within [0, 1] generated anew for each value of *j*. $x''_{i,j}$ is the parent vector, $x'_{i,j}$ the mutant vector and $x_{i,j}$ the trial vector.

5 Simulation Results

Proposed algorithm has been applied to solve the reactive power planning problem. Here, IEEE 30 bus network is considered to verify its viability and effectiveness. This network consists of 30 buses and 41 branches from which we select 13 control variables for our reactive power planning problem [24]. These control variables are as follows; 5 numbers generator buses, 4 numbers on-load tap changing transformer coefficients. Again 4 numbers of buses have compensating shunt devices used also as control variable. The algorithm has been written in house in MATLAB and run on a 2.4GHz, 1GB RAM PC. The solution obtained by the proposed method has been compared with evolutionary programming (EP) and Broyden's method [24] etc.

In this paper one year's energy loss cost is used to assess the opportunity of installing the VAR sources. The initial value of real power generation is 2.89388p.u and the load during the period is 2.834p.u. The base value of transmission line loss is 0.05988 p.u. Table 1 gives the parameter values for simulation which are selected by trial and error method.

Table 1. Simulation parameters used for proposed algorithm

| | |
|------------------------------|-------------|
| Parameters | IEEE 30-bus |
| Population size | 20 |
| Maximum number of generation | 100 |
| Crossover rate | 0.2 |

The details about the maximum and minimum limits of the control variables are taken from elsewhere [24]. The optimal values of the control variables and power loss obtained from the proposed algorithm are presented in the Tables 2. The insertion of crossover operator helps to accelerate the optimization giving better convergence. It is seen from the Tables that, after iterating 100 generations the Improved CA is more

capable to reduce the cost involved in reactive power planning as compared to CA. The transmission loss is obtained as 0.0145 p.u. by proposed method. Whereas by CA transmission loss was found to be reduced to 0.0189 p.u. from the base value of 0.05988 p.u.

Table 2. Simulation Result for IEEE 30-bus System for different approaches

| Variable | Results | | |
|--------------------|---------------------|-----------|---------|
| | CA | Hybrid CA | |
| Generator Voltages | V ₁ | 1.0500 | 1.0500 |
| | V ₂ | 1.0478 | 1.0438 |
| | V ₅ | 1.0272 | 1.0343 |
| | V ₈ | 1.0234 | 1.0497 |
| | V ₁₁ | 1.0966 | 1.0348 |
| | V ₁₃ | 0.9993 | 1.0387 |
| Transformer Taps | T _{C6-9} | 1.0128 | 1.0140 |
| | T _{C6-10} | 0.9663 | 0.9853 |
| | T _{C4-12} | 1.0067 | 1.0328 |
| | T _{C28-27} | 0.9520 | 0.9578 |
| Shunt Compensation | Q _{C6} | 4.5426 | 26.1758 |
| | Q _{C17} | 11.7769 | 3.9712 |
| | Q _{C18} | 20.4522 | 14.8851 |
| | Q _{C27} | 8.5749 | 3.3877 |
| Power Loss | P _{loss} | 0.0189 | 0.0145 |

Based on the system losses, the power saving methodology can be derived by which expenses are also minimized. The real power savings, annual cost savings and the total costs are calculated by using the simulation results as follows for both the approaches,

1) *With CA*

$$P_{save} \% = \frac{P_{loss}^{init} - P_{loss}^{opt}}{P_{loss}^{init}} \times 100 = \frac{0.05988 - 0.0189}{0.05988} \times 100 = 68.43\%$$

$$\begin{aligned} W_C^{save} &= hd_1 (P_{loss}^{init} - P_{loss}^{opt}) \\ &= 0.06 \times 8760 \times (0.05988 - 0.0189) \times 10^5 \\ &= \$2,153,908 \end{aligned}$$

$$F_C = I_C + W_C = 1,364,400 + 9,94,850 = \$2,359,200$$

2) *With Improved CA*

$$P_{save} \% = \frac{P_{loss}^{init} - P_{loss}^{opt}}{P_{loss}^{init}} \times 100 = \frac{0.05988 - 0.0145}{0.05988} \times 100 = 75.78\%$$

$$\begin{aligned} W_C^{save} &= hd_l (P_{loss}^{init} - P_{loss}^{opt}) \\ &= 0.06 \times 8760 \times (0.05988 - 0.0145) \times 10^5 \\ &= \$2,385,172 \end{aligned}$$

$$F_C = I_C + W_C = 1,456,600 + 7,61,980 = \$2,218,600$$

Table 3. Comparison of the Simulation Result with other Algorithm

| Method | W_C^{save} (\$) | F_C |
|--------------|-------------------|-----------|
| EP [24] | 538,740 | 2,608,500 |
| Broyden [24] | 132,451 | 4,013,280 |
| CA | 2,153,908 | 2,359,200 |
| Hybrid CA | 2,385,172 | 2,218,600 |

Table 3 compares the results obtained by between different algorithms. It is seen that the method of hybrid Cultural Algorithm shows the efficacy of the proposed algorithm. The comparison shows that the improved cultural algorithm is capable of producing lower cost over the other methods which is the objective of the RPP problem. The feasible solution for the total cost involved in reactive power planning in each generation is plotted in the Fig.1 which shows the effectiveness of the proposed method over the simple cultural algorithm as the search process of the proposed method converges rapidly towards the optimal solution. The bus voltages are depicted in Fig.2 which shows that the values of bus voltages obtained by hybrid cultural algorithm lie

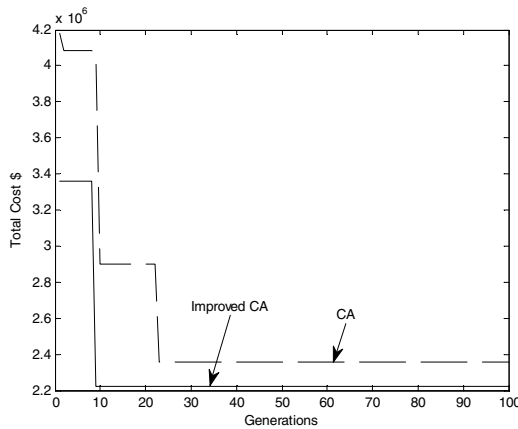


Fig. 1. Convergence characteristic for optimal cost

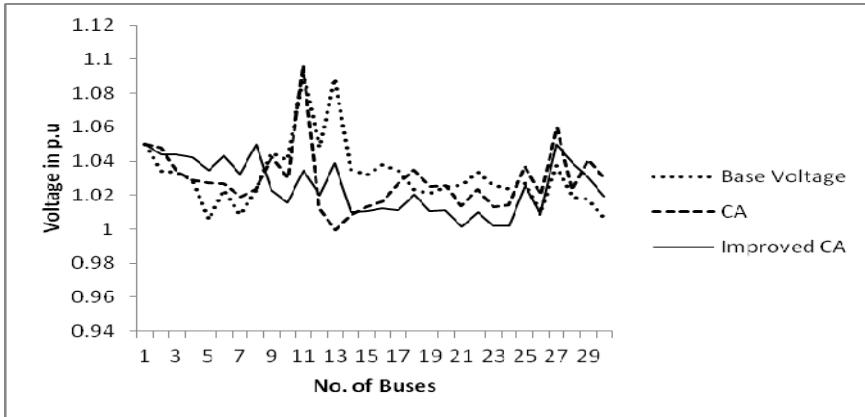


Fig. 2. Bus voltage profile of the test system

in maximum and minimum range smoothly. Whereas, in base value and the values obtained from simple cultural algorithm shows sharpness in bus voltages. Thus we can say that the voltage profile achieved from Hybrid CA is more stable than the base value as well as than the other method. Hence the optimum solution technique achieved from the proposed algorithm gives better operation for the problem in hand.

6 Conclusions

This paper has proposed a different approach of cultural algorithm for the reactive power planning problem in power systems. The simulation results have shown that the proposed method is better than other methods in terms of the convergence characteristics and accuracy. Not only is there active power loss minimization, the reactive power cost and the quality of voltage transferred to the customers are also optimized in this method. From this limited comparative study, it can be concluded that the applied algorithm can be effectively used to solve smooth as well as non-smooth constrained RPP problems. In future, efforts will be made to incorporate more realistic constraints to the problem structure and the practical large sized problems would be attempted by the proposed methodology.

References

1. Sachdeva, S.S., Billington, R.: Optimum network VAR planning by nonlinear programming. *IEEE Trans Power Apparatus Systems*, PAS-92, 1217–1225 (1973)
2. Quintana, V.H., Satos-nieto, M.: Reactive power dispatch by successive quadratic programming. *IEEE Trans. Energy Convers* 4(3), 425–435 (1989)
3. Aoki, K., Fan, M., Nishikori, A., Optimal, V.A.R.: planning by approximation method for recursive mixed integer linear programming. *IEEE Trans. Power Systems* 3(4), 1741–1747 (1988)
4. Deeb, N., Shahidehpour, S.M.: Linear reactive power optimization in a large power network using the decomposition approach. *IEEE Trans. on Pwr. Syst.* 5(2), 428–438 (1990)

5. Da Costa, G.R.M.: Optimal reactive dispatch through primal-dual method. *IEEE Trans. on Power Systems* 12(2), 669–674 (1997)
6. Iba, K.: Reactive power optimization by genetic algorithm. *IEEE Trans. on Pwr. Syst.* 9(2), 685–692 (1994)
7. Wu, Q.H., Ma, J.T.: Power system optimal reactive power dispatch using evolutionary programming. *IEEE Trans. Power Syst.* 10(3), 1243–1249 (1995)
8. Lee, K.Y., Bai, X., Park, Y.M.: Optimization method for reactive power planning by using a modified simple genetic algorithm. *IEEE Trans. Power Syst.* 10(4), 1843–1850 (1995)
9. Liu, M., Ma, L., Zhang, J.: GA/SA/TS Hybrid Algorithms for Reactive Power Optimization. *Power Eng. Soc. Summer Meeting 1*, 245–249 (2000)
10. Mandal, K.K., Chakraborty, N.: Optimal Capacitor Placement in Distribution Systems Using New Improved Particle Swarm Optimization Technique. *International Journal of Applied Engineering Research* 6(5), 1182–1188 (2011)
11. Esmin, A.A.A., Lambert-Torres, G., de Souza, A.C.Z.: A hybrid particle swarm optimization applied to loss power minimization. *IEEE Trans. Power Syst.* 20(2), 859–866 (2005)
12. Abbasy, A., Tabatabal, I., Hosseini, S.H.: Optimal Reactive Power Dispatch in Electricity Markets Using A Multiagent-Based Differential Evolution Algorithm, Setubal, Portugal, *POWERENG* (2007)
13. Ghoshal, S.P., Chatterjee, A., Mukherjee, V.: Bio-inspired fuzzy logic based tuning of Power system stabilizer. *Expert. Syst. Appl.* 36(5), 9281–9292 (2009)
14. Pothiya, S., Ngamroo, I., Kongprawechnon, W.: Ant colony optimisation for economic dispatch problem with non-smooth cost functions. *Int. J. Elect. Power Energy Syst.* 32(5), 478–487 (2010)
15. Krishnanand, K.R., Hasani, S.M.F., Panigrahi, B.K., Panda, S.K.: Optimal Power Flow Solution Using Self-Evolving Brain-Storming Inclusive Teaching-Learning-Based Algorithm. In: Tan, Y., Shi, Y., Mo, H. (eds.) *ICSI 2013, Part I. LNCS*, vol. 7928, pp. 338–345. Springer, Heidelberg (2013)
16. Antunes, C.H., Oliveira, E., Lima, P.: A multi-objective GRASP procedure for reactive power compensation planning. In: *Optimization and Engineering* (September 2013)
17. Reynolds, R.G.: An Adaptive Computer Model of the Evolution of Agriculture for Hunter-Gatherers in the Valley of Oaxaca Mexico. Ph.D. Dissertation, Department of Computer Science, University of Michigan (1979)
18. Reynolds, R.G.: An Introduction to Cultural Algorithms. In: *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, pp. 131–139. World Scientific Publishing (1994)
19. Liu, J., Gao, H., Zhang, J., Dai, B.: Urban power network substation optimal planning based on Geographic Cultural Algorithm. In: *The 8th Int. Conf. Power Engineering*, pp. 500–504 (2007)
20. Yuan, X., Yuan, Y.: Application of Cultural Algorithm to generation scheduling of hydro-thermal systems. *Energy Conversion and Management* 47(15-18), 2192–2201 (2006)
21. Seifi, A.R.: A new Hybrid Optimization method for optimum distribution capacitor planning. In: *Modern Applied Science, CCSE*, vol. 3(4), pp. 196–202 (April 2009)
22. Bhattacharya, B., Mandal Kamal, K., Chakraborty, N.: A Multiobjective Optimization Based on Cultural Algorithm for Economic Dispatch with Environmental Constraints. *International Journal of Scientific & Engineering Research* 3(6), 1–9 (2012)
23. Bhattacharya, B., Mandal, K.K., Chakraborty, N.: Reactive Power Optimization using Hybrid Cultural Algorithm. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Nanda, P.K. (eds.) *SEMCCO 2012. LNCS*, vol. 7677, pp. 106–115. Springer, Heidelberg (2012)
24. Lai, L.L., Ma, J.T.: Application of Evolutionary Programming to Reactive Power Planning – Comparison with Nonlinear Programming Approach. *IEEE Trans. Power Systems PAS-12*(1), 198–206 (1997)

BFO-RLDA: A New Classification Scheme for Face Images Using Probabilistic Reasoning Model

Lingraj Dora¹, Sanjay Agrawal², and Rutuparna Panda³

¹ Electrical & Electronics Engineering Department

^{2,3} Electronics & Telecommunication Engineering Department,

VSS University of Technology, Burla, Odisha, India

lingraj02uce157ster@gmail.com,

agrawals.72@gmail.com,

r_ppanda@yahoo.co.in

Abstract. Regularized-LDA (R-LDA) is a LDA-based method used for finding the nonzero eigenvalues and the corresponding eigenvectors. This paper presents a RLDA-based classification of face images which uses bacteria foraging optimization (BFO) algorithm for selecting the optimal discriminating features. The optimal features are then used by probabilistic reasoning model for classification of unknown face images. The ORL and the UMIST databases are used for experiment to demonstrate the performance of our proposed method. It is observed that our proposed method outperforms the existing method.

Keywords: Face Classification, Face Recognition, Linear Discriminant Analysis (LDA), Regularized-LDA (R-LDA), Bacteria Foraging Optimization (BFO), Probabilistic Reasoning Model (PRM).

1 Introduction

Face classification is the task of identifying the class of a person from various classes, where a class contains the images of same person. Face classification can be carried out by projecting the face images into a low dimensional feature space. Then an unknown face image is classified by comparing its position in the feature space. Face classification can be effectively applied in the field of face recognition (FR), person identification and surveillance task. There are number of algorithms and methods available in the literature to carry out the task of face recognition and classification. They are broadly divided into two types: Appearance-based methods and Feature-based methods. Among the various types of approaches available in the literature, the most successful one is the appearance-based method [1-2], which processes the image as two-dimensional holistic pattern. In this type of subspace analysis, an unknown face image is projected into a lower dimensional space called feature space. Then the unknown face image is classified by comparing its position in the feature space of known face images.

Principal Component Analysis (PCA) [1] and Linear Discriminant Analysis (LDA) [2-3] has been the two popular techniques of the appearance-based method employed in face classification and recognition [1], [2].

PCA has been effectively used in the Eigenface method for face classification as proposed by Turk et al. The method uses PCA for linearly projecting the high dimensional face images into a low dimensional feature space. The basic principle of PCA is to find the eigenvectors (principal components) of the high dimensional face images so as to obtain maximum variance direction in the original image space corresponding to the basis vector [1-2].

Linear Discriminant Analysis is a class specific method used for feature extraction and dimension reduction and has been proven to give better recognition rates than PCA [2]. The basic idea of LDA is to maximize the ratio of between class scatter matrix to the within class scatter matrix so that between class scatter is maximized and the within class scatter matrix is minimized, and thereby maximizing the discriminatory features between classes of samples. But the basic LDA suffers from the singularity problem which arises when all the scatter matrices are singular [2-5]. To avoid the problem of singularity, Belhumer et al. [2] proposed a method called Fisher-Linear Discriminant analysis (F-LDA) which uses principal component analysis (PCA) [1] as a pre-processing step in the LDA [2]. The F-LDA is also not sensitive to variation in lighting condition and facial expression and the method give a lesser error rate as compare to Eigenface method.

Another problem which generally persists in LDA-based methods is called the Small Sample Size (SSS) [3-7]. As proposed by Raudys et al. this problem arises in LDA-based methods while doing high dimensional pattern recognition task like FR, where the number of training samples for each subject is smaller than the dimensionality of the samples [6]. To overcome such kind of problem, various types of LDA-based methods are used like PCA followed by LDA [2], Direct-LDA (DLDA) [4], Regularized-LDA (R-LDA) [5, 7].

Belhumeur et al. [2] uses F-LDA method, where PCA as a pre-processing step in LDA to reduce the dimension and thereby removing the null space (zero or small eigenvalue) of within class scatter matrix. But sometimes the discarded null space may also contain significant information. Thus D-LDA method is presented by Chen et al. in 2000 and Yu et al. in 2001 which processes the image directly on the high dimensional space. But the D-LDA method suffers from the problem of high variance while computing the null space of within class scatter matrix as presented in the paper of Lu et al [5]. As proposed by Lu et al. this problem of high variance can be eliminated by introducing a new variant of LDA-based method called R-LDA. The R-LDA includes a regularization parameter (η) which is used to stabilize the high variance with a trade-off with the bias. At the same time regularization parameter can also be used to stabilize the smaller eigenvalues. The R-LDA method has been well used in FR task [5]. Dora et al. has presented an improved method called R-LDA using PRM [8] which can also be used in FR task, and the results shows an improved recognition rate as compared to the R-LDA method. The method uses R-LDA for feature extraction and then PRM is used for the classification of unknown test image.

In all the above methods, the basic principle is to obtain the first 20 or 30 eigenvectors corresponding to the largest eigenvalues which contains significant information as presented in the paper of Panda et al. However, the eigenvectors having smaller eigenvalues may also contain some significant information. And there is no such

criterion given for the selection of significant eigenvectors. So for the selection of optimal eigenvectors optimization algorithms like Genetic algorithm (GA) [9-10] and Bacteria Foraging Optimization algorithm (BFO) [12-13] can be used as proposed by Zheng et al. and Panda et al. Evolutionary pursuit is a method which uses GA for selection of optimal principal components from PCA and then uses the optimal components for FR [9]. GA-Fisher is another example of such approach. Here GA is used for the selection of optimal principal components from PCA and then the optimal components are used to project the image to a lower dimensional subspace, thereby eliminating the singularity problem and then the projected images are used in Fisher-faces method for FR [10].

BFO algorithm [12-13] has also been successfully used to find the optimal vectors from PCA using the same fitness function as used in GA-Fisher method. Then after the selection of optimal principal components, a whitening method followed by LDA is used for FR. This method of FR is called as EBFS-Fisher and it is demonstrated that it outperforms the GA-Fisher method [11]. The above two methods, GA-Fisher and EBFS-Fisher are well applied to FR task. But previously, no work has been reported in the literature which applies BFO directly into the LDA-based method for face classification. This has motivated us to use BFO in R-LDA for classification of face images. In this method, instead of taking the first eigenvectors corresponding to the largest eigenvalues, we have used the BFO in the R-LDA space to find the p optimal vectors from $m(\leq K - 1)$ eigenvectors. The method uses a new proposed fitness function to find the optimal vectors and the results reveal that the optimal vectors give better classification rate as compared to the simple R-LDA method. This work may add a new dimension to this area of research.

In this paper, we propose a face classification method which uses BFO-RLDA for classification of face images using probabilistic reasoning model (PRM). The basic assumption made here is that the trained database contains at least five images of same subject to be classified. The BFO-RLDA is used for projecting the images to a low dimensional feature space. The RLDA is used for calculating the discriminating eigenvectors from the high dimensional image data in such a way that both the singularity and the SSS problems are eliminated. BFO which follows the RLDA is used for the selection of optimal eigenvectors. A new fitness function is defined here for BFO to select the optimal eigenvectors. At last the PRM method is used for the classification of unknown face images. To show the effectiveness of the proposed method, the ORL and the UMIST face databases are used. The evaluation results also show that with less number of features we are able to get better classification rates.

The paper is organized as follows: Section 1 contains the introduction and literature survey on various methods used in face classification and recognition. Section 2 contains a brief explanation about R-LDA method. Section 3 contains our proposed BFO-RLDA method. The results and discussions are presented in Section 4. Conclusion is presented in Section 5.

2 Regularized-LDA (R-LDA) Method

Let $X = [x_1, x_2, \dots, x_N]$ be a set of N number of training face images. Each image x_i is of size $n = wh \times l$, which is obtained by lexicographic ordering of the pixels in the image. Here, w, h represent size of an image. In the training data set, each image x_i belongs to one of K classes. Let C_i be the number of images in class i and the mean of i^{th} class is given by $M_i = (1/C_i) \sum_{j=1}^{C_i} X_{ij}$. Now the between-class scatter matrix is given by:

$$S_B = \sum_{i=1}^K \varphi_{B,i} \varphi_{B,i}' = \varphi_B \varphi_B' \quad (1)$$

And the within-class scatter matrix is given by:

$$S_W = \frac{1}{N} \sum_{i=1}^K \sum_{j=1}^{C_i} (X_{ij} - M_i)(X_{ij} - M_i)' \quad (2)$$

Where, $\varphi_B = [\varphi_{B,1}, \varphi_{B,2}, \dots, \varphi_{B,K}]$, $\varphi_{B,i} = (C_i/T)^{1/2} (M_i - M)$, M is the total mean of the training images and t represents the transpose operator. The R-LDA method projects the images having high dimension (n) to a subspace having lower dimension m by using Γ , where $\Gamma = [\Gamma_1, \Gamma_2, \dots, \Gamma_m]$ represents the set of basis vectors such that $m \ll N$. The set of basis vectors is used to reduce the high dimension of images and to project the images to a low dimensional feature space. In the R-LDA method, the basis vectors are calculated by taking eigenvectors having largest eigenvalues [5, 7]. But sometimes the eigenvectors having smaller eigenvalue also contains significant information. So an optimal search method like GA and BFO can be used to select the significant eigenvectors [10-11]. Here BFO algorithm is used for the selection of optimal p number of eigenvectors from R-LDA method in the high dimensional space and thus an optimal set of basis vectors are obtained such that $p \ll m$. By this technique, more discrimination is achieved in the feature space.

3 Proposed BFO-RLDA Method

In the proposed BFO-RLDA method, we calculate the between class matrix (S_B) and the within class scatter matrix (S_W) from the training data set of N images by using (1) and (2). The eigenvectors of (S_B) is calculated in an indirect way from $\varphi_B^t \varphi_B$ matrix, having a size of $K \times K$. Consider $V = [\nu_1, \nu_2, \dots, \nu_m]$, where the columns of V

represent the first m most significant eigenvectors of $\varphi_B^t \varphi_B$ having non-zero eigenvalues with $m \leq K - 1$. Now the m most significant eigenvectors and eigenvalues of S_B are given as $E = \varphi_B V$ and $\Delta_B = E^T S_B E$. Let $H = E \Delta_B^{-1}$ and $P = [P_1, P_2, \dots, P_m]$, where P contains the m eigenvectors of $H^t S_w H$. From the m eigenvectors, the first p eigenvectors are selected having largest eigenvalues for calculation of basis vectors (Γ). But, sometimes the eigenvectors having smaller eigenvalues also contain significant information. Here, we have used BFO algorithm for selecting the p most optimal or effective eigenvectors. The BFO algorithm uses a swarm of bacteria in the search space to find the optimum global solution. A m dimension bacterium is used here to represent the contribution of each eigenvector. A new fitness function for BFO is also proposed here to select the p most optimal eigenvectors from m eigenvectors. Let x be a face image and $\bar{x} = \Gamma^t x$ be the face image in the low-dimension space. The total mean of all the images is given by $\bar{M} = \Gamma^t M$ and $\bar{M}_i = \Gamma^t M_i$ represents the mean of each class. Let

$$F_b = \sqrt{\sum_{i=1}^K (\bar{M}_i - \bar{M})^t (\bar{M}_i - \bar{M})} \tag{3}$$

and

$$F_w = \sum_i \sum_j (\bar{x} - \bar{M}_i)^t (\bar{x} - \bar{M}_j) \tag{4}$$

Now the fitness function $J(\theta)$ used in BFO algorithm is given by:

$$F = \frac{F_b}{F_w} \tag{5}$$

and

$$J = \frac{1}{F} \tag{6}$$

Finally the optimal eigenvectors are selected as:

$$\Gamma_{opt} = \arg \max_p \{ \Gamma \} \tag{7}$$

Where Γ_{opt} represent the optimal set of eigenvectors for which the fitness function is $J(\theta)$ maximized. The following steps are used in the algorithm for feature extraction:

1. At first, $V = [\nu_1, \nu_2, \dots, \nu_m]$, the first m most significant eigenvectors of $\varphi_B' \varphi_B$ having non-zero eigenvalues are calculated where $m \leq K - 1$. Then the eigenvectors and eigenvalues of S_B are calculated as: $E = \varphi_B V$ and $\Delta_B = E^T S_B E$.
2. Then $P = [P_1, P_2, \dots, P_m]$, the eigenvectors of $H^T S_W H$ having eigenvalues (Δ_m) are calculated. Where, $H = E \Delta_B^{-1}$ and S_W is calculated from (2).
3. The BFO algorithm is used to select p most optimal eigenvectors from P . Let $P = [P_1, P_2, \dots, P_p]$ be the selected optimal eigenvectors from BFO algorithm having eigenvalues (Δ_p) such that $p \ll m$.
4. Now using the obtained p optimal eigenvectors, the significant basis vectors of R-LDA are calculated as: $\Psi = H P_p (\eta I + \Delta_p)^{-1/2}$, where I represents the identity matrix and η represents the regularization parameter. In this paper value of η taken is 10^{-3} . Then the optimal most significant discriminating features (Y) of the face images (X) are calculated using Ψ , as $Y = \Psi^t X$.



Fig. 1. Example of training images of two subjects from ORL database. Here five images of same subject are called a class.



Fig. 2. Example of training images of two subjects from UMIST database. Here eight images of each subject are called a class.

After the calculation of optimal features (Y) of the face images from the BFO-RLDA method, the PRM method which is a measure of class separability is used to classify the unknown face images. The PRM method is defined as a bayes linear classifier

under the assumption that the within class covariance matrices are identical and diagonal [7, 14-15]. Here MAP (Maximum *A Posteriori*) classification rule is used which uses PRM for classification. The MAP classification rule classifies the image feature vector (Y) into class C_k as follows:

$$\sum_{i=1}^K \frac{(Y_i - \mu_{ki})^2}{\rho_i^2} = \min_j \left\{ \sum_{i=1}^K \frac{(Y_i - \mu_{ji})^2}{\rho_i^2} \right\} \rightarrow Y \in C_k \quad (8)$$

where, the component ρ_i^2 is estimated in one-dimensional feature subspace as:

$$\rho_i^2 = \frac{1}{K} \sum_{k=1}^K \left\{ \frac{1}{N_k - 1} \sum_{j=1}^{N_k} (y_{ji}^{(k)} - \mu_{ki})^2 \right\} \quad (9)$$

where, $y_{ji}^{(k)}$ is the i^{th} element of the sample $Y_j^{(K)}$ of face image that belongs to class C_k and μ_{ki} is the i^{th} element of μ_k (mean of each class in feature space). K is the number of classes and C_k is the number of images in each class.

4 Results and Discussions

To demonstrate the effectiveness of the proposed method two face databases: the ORL and the UMIST databases are taken. The ORL face database consists of 400 sample face images of 40 subjects with each subject having 10 images and the UMIST face database consists of 1012 sample face images of 20 subjects. Two training datasets are made by randomly taking 5 images of each subject from ORL and 8 images of each subject from UMIST database. Examples of some training images are shown in fig. 1 and fig. 2. At first, each images in the two databases are normalized to 70×90 , resulting in a dimension of $n = 6300$. The parameter setting for the BFO-RLDA algorithm is given as: $S = 8$, $N_c = 5$, $N_S = 4$, $N_{re} = 4$, $N_{ed} = 2$, $d_{attract} = 0.01$, $w_{attract} = 0.2$, $h_{repellent} = 0.01$, $w_{repellent} = 10$, $C(i) = 0.066$.

Fig. 3 and Fig. 4 show the discriminatory power of BFO-RLDA and RLDA using PRM (RLDA-PRM) methods. Here, 25 images of 5 different subjects from ORL database are plotted in Fig. 3 by taking two significant features from both the methods in the feature space. Similarly, Fig. 4 shows the plot of 40 images of 5 different subjects from UMIST database. Five different symbols (+, •, *, ◊, ▽) are used in the plot to represent the images of 5 subjects and a group of each symbol denotes one class.

Fig. 3 and Fig. 4 reveal that the face images of different subjects are well separated and improved clustering is achieved by our proposed method.

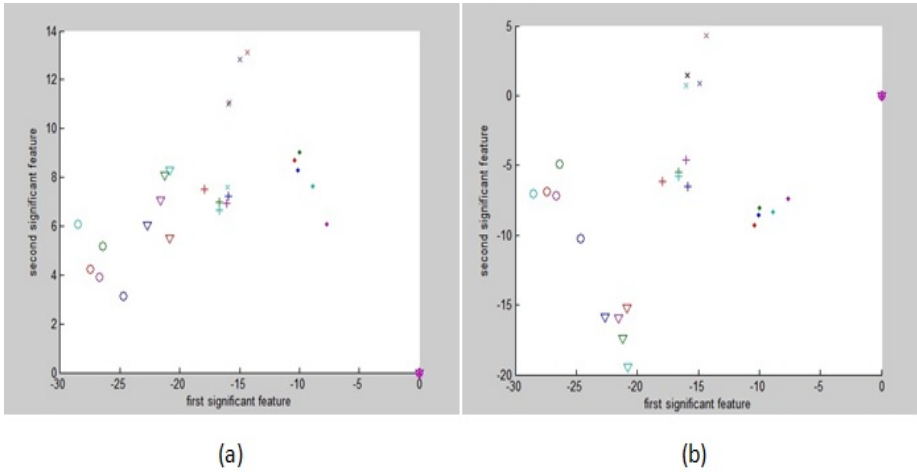


Fig. 3. Classification result with ORL database. (a) Using RLDA-PRM. (b) Using BFO-RLDA method.

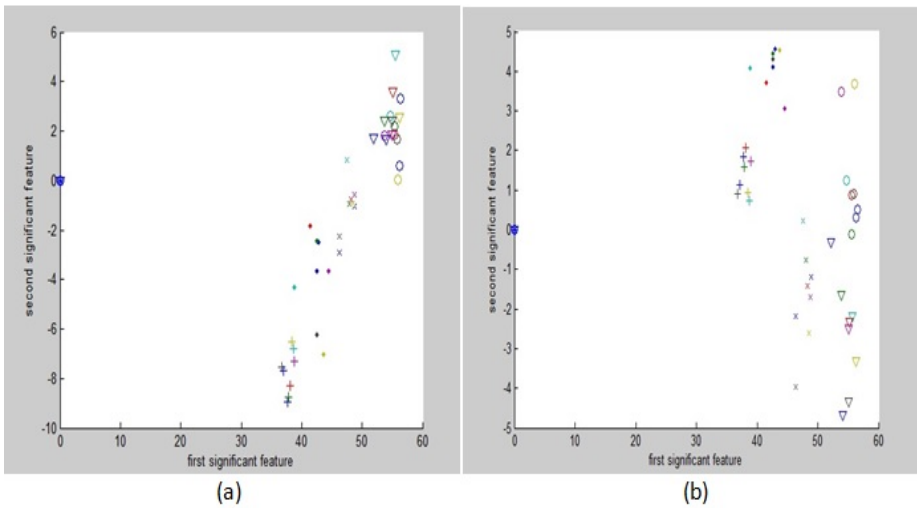


Fig. 4. Classification results with UMIST database. (a) Using RLDA-PRM method. (b) Using BFO-RLDA method.

Table 1 shows the classification rates obtained from the RLDA-PRM and BFO-RLDA methods. From the table it is observed that the recognition rates are increased by the BFO-RLDA method using less number of features.

Table 1. Classification Rates Using ORL and UMIST databases

| Name of Database | No. of features | Methods | |
|------------------|-----------------|----------|----------|
| | | RLDA-PRM | BFO-RLDA |
| ORL | 8 | 90.5% | 93% |
| | 16 | 96% | 97% |
| | 20 | 97% | 98.5% |
| UMIST | 4 | 93.78% | 94.14% |
| | 6 | 95.78% | 96.13% |
| | 10 | 97.89% | 98.12% |

Table 2. Classification of face images in ORL and UMIST databases


| Test Image |  |  |  |  |  |
|---------------------|---|---|---|---|---|
| Actual Class | 11 | 14 | 26 | 2 | 7 |
| Class from BFO-RLDA | 11 | 14 | 10 | 2 | 12 |

Table 2 shows the classification of test images obtain from the ORL and UMIST databases using BFO-RLDA method. When an unknown test face image is given as input to the BFO-RLDA method than the method give a class number as output into which the test image belongs. By using this method many of the test face images are properly classified to their actual class and for some few test images misclassification occurs.

5 Conclusion

In this paper we proposed a BFO-RLDA method for face classification. The method uses BFO directly in the RLDA method for selecting the optimal eigenvectors. Then these optimal set of eigenvectors are used to project the images to feature space and then used in PRM for classification. Here two face databases ORL and UMIST are used to demonstrate the efficiency of the propose method. And from the results it is observed that BFO-RLDA outperforms RLDA-PRM using less number of features. The algorithm will be validated by taking less number of images for training, using FERET database. The idea can also be extended for face recognition.

References

1. Turk, M., Pentland, A.P.: Eigenfaces for Recognition. *Journal of Cognitive Neuroscience* 3(1), 71–86 (1991)
2. Belhumer, P.N., Hespanha, J.P., Krieman, D.J.: Eigenfaces vs Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis Machine Intelligence* 19, 711–720 (1997)

3. Juwei, L., Plataniotis, K.N., Venetsanopoulos, A.N.: Face Recognition Using LDA-Based Algorithms. *IEEE Transactions on Neural Networks* 14(1), 195–200 (2003)
4. Yu, H., Yang, J.: A Direct LDA Algorithm for High-Dimensional Data with Application to Face Recognition. *Pattern Recognition* 34, 2067–2070 (2001)
5. Juwei, L., Plataniotis, K.N., Venetsanopoulos, A.N.: Regularization Studies of Linear Discriminant Analysis in Small Sample Size Scenarios with Application to Face Recognition. *Pattern Recognition Letters* 26(2), 181–191 (2005)
6. Raudys, S.J., Jain, A.K.: Small Sample Size Effects in Statistical Pattern Recognition: Recommendation for Practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(3), 252–264 (1991)
7. Chen, L.F., Mark Liao, H.Y., Ko, M.T., Lin, J.C., Yu, G.J.: A New LDA-Based Face Recognition System Which can Solve the Small Sample Size Problem. *Pattern Recognition* 33, 1713–1726 (2000)
8. Dora, L., Rath, N.P.: Face Recognition by Regularized-LDA Using PRM. In: *IEEE International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 140–145 (2010)
9. Liu, C., Wechsler, H.: Evolutionary Pursuit and its Application to Face Recognition. *IEEE Transactions on Pattern Analysis Machine Intelligence* 22, 570–582 (2000)
10. Zheng, W.S., Lai, J.H., Yuen, P.C.: GA-Fisher: A New LDA-Based Face Recognition Algorithm with Selection of Principal Components. *IEEE Transactions on Systems Man Cybernetics, Part B* 35(5), 1065–1078 (2005)
11. Panda, R., Naik, M.K.: EBFS-Fisher: An Efficient Algorithm for LDA-Based Face Recognition. In: *World Congress on Nature & Biologically Inspired Computing*, pp. 1041–1046 (2009)
12. Passino, K.M.: Biomimicry of Bacteria Foraging for Distributed Optimization and Control. *IEEE Control Systems Magazine*, 52–67 (2002)
13. Dasgupta, S., Das, S., Abraham, A., Biswas, A.: Adaptive Computational Chemotaxis in Bacterial Foraging Optimization: An Analysis. *IEEE Transactions on Evolutionary Computation* 13(4), 919–941 (2009)
14. Liu, C., Wechsler, H.: Probabilistic Reasoning Models for Face Recognition. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California (1998)
15. Liu, C., Wechsler, H.: Independent Component Analysis of Gabor Features for Face Recognition. *IEEE Transactions on Neural Networks* 14(4), 919–928 (2003)
16. ORL Face Database, <http://www.com-orl.co.uk/facedatabse.html>
17. UMIST Face Database, <http://images.ee.umist.ac.uk/danny/database.html>

Optimal Stable IIR Low Pass Filter Design Using Modified Firefly Algorithm

Suman K. Saha¹, Rajib Kar¹, Durbadal Mandal¹, and Sakti Prasad Ghoshal²

¹Department of Electronics and Comm. Engineering

²Department of Electrical Engineering,

National Institute of Technology, Durgapur,

West Bengal, India

{namus.ahas, spghoshalnitdgp}@gmail.com

Abstract. In this paper, a recently proposed global heuristic search optimization technique, namely, Modified Firefly Algorithm (MFFA) is considered for the design of the 8th order infinite impulse response (IIR) low pass (LP) digital filter. This modified version of FFA is considered to achieve quality output response by means of properly tuned control parameters over conventional Firefly Algorithm (FFA). Newly defined randomization parameter and modification in updating formula in MFFA makes it a perfect search tool in multidimensional search space. With this approach better exploration and exploitation are achieved, which have resulted in faster convergence to near global optimal solution. The performance of the proposed MFFA based approach is compared to the performances of some well accepted evolutionary algorithms such as particle swarm optimization (PSO) and real coded genetic algorithm (RGA). From the simulation study it is established that the proposed optimization technique MFFA outperforms RGA and PSO, not only in the accuracy of the designed filter but also in the convergence speed and the solution quality, i.e., the stop band attenuation, transition width, pass band and stop band ripples.

1 Introduction

In the signal processing system, filtering holds a significant position which is involved with manipulation by modifying, reshaping or transforming the spectrum of signal. Fundamentally, a filter operates on frequency domain to permit certain band of frequencies to pass through and attenuates others. The frequency at which such phenomenon happens is a design dependent parameter called cut-off frequency. This sort of frequency discrimination is of prime importance due to mixing of information carrying signal with noise. There are different sources of noise either created by nature or man-made effects. According to its frequency domain characteristics and source of generation, signals are mostly contaminated with thermal noise, shot noise, avalanche noise, flicker noise etc.

Most of the filters can be implemented with discrete components like resistor, capacitor, inductor and operational amplifiers when the input signal is continuous

function of time and hence these are called analog filters. On the other hand a digital filter performs mathematical operations on a sampled, discrete time signal to reduce or enhance the desired features of the applied signal. Analog filters are replaced by digital filters due to their wide range of applications and superior performance. The advantages of digital filters over analog filters are small physical size, high accuracy, reliability and immune to component tolerance sensitivity [1].

Digital filters are of two types: finite impulse response (FIR) and infinite impulse response (IIR) filter. The order of the IIR filter is lower than that of the FIR filter for the same design specifications such as cut-off frequencies, pass band and stop band ripples and stop band attenuation. Hence, lesser number of delay elements and multipliers are required for hardware implementation and also lower computational time is required for software realization for IIR filter design [2].

Minimization of an objective function (typically the mean square error between desired response and estimated filter output) is often performed by gradient based iterative search algorithms. However, when the error surface (objective function) is multimodal and/or non-smooth, gradient-based optimization methods often cannot succeed in converging to the global minimum.

So, meta-heuristic evolutionary methods have been employed in the design of digital filters to design with better parameter control and to better approximate the ideal filter. Evolutionary optimization methods that require no gradient and can achieve a near global optimal solution offer considerable advantages in solving these multi-modal objective functions in digital filter design problem.

Different heuristic search techniques are reported in the literature. These are GA [3-4], Seeker optimization Algorithm (SOA) [5], orthogonal genetic algorithm (OGA) [6], hybrid Taguchi GA (TGA) [7], Tabu search [8], Simulated Annealing (SA) [9], Bee Colony Algorithm (BCA) [10], Differential Evolution (DE) [11], Cat swarm Optimization [12], Artificial Immune Algorithm [13], particle swarm optimization (PSO) [14-16], Gravitational search algorithm (GSA) [17-18], Opposition based BAT algorithm (OBA) [19], Firefly algorithm (FFA) [20-25] etc.

The approach detailed in this paper takes advantage of the power of the stochastic global optimization technique called modified Firefly algorithm (MFFA). Although the algorithm is adequate for applications in any kind of parameterized filters, the authors have chosen to focus on real-coefficient IIR filters. The basic Firefly algorithm is very efficient. It is suitable for parallel implementation because different fireflies can work almost independently. But it is observed from the simulation results that the solutions are still changing as the optima are approaching. To improve the solution quality, randomness is reduced so that the algorithm could converge to the optimum more quickly [26]. Apart from normal FFA the modifications considered in MFFA are as follows. In FFA, randomization parameter is a random number but in MFFA it is a gradually decreasing function of iteration cycle and in position updating formula, position of the group best firefly is taken into consideration for the calculation of new position of any firefly. With these modifications the solution obtained is much close to the global optimal solution with less number of iteration cycles.

2 Low Pass IIR Filter Design

This section presents the design strategy of IIR filter based on MFFA. The input-output relation is governed by the following difference equation [2]:

$$y(p) + \sum_{k=1}^n a_k y(p-k) = \sum_{k=0}^m b_k x(p-k) \quad (1)$$

where $x(p)$ and $y(p)$ are the filter's input and output, respectively, and $n(\geq m)$ is the filter's order. The transfer function of IIR filter with the assumption $a_0 = 1$ is expressed as in (2).

$$H(z) = \frac{\sum_{k=0}^m b_k z^{-k}}{1 + \sum_{k=1}^n a_k z^{-k}} \quad (2)$$

Let $z = e^{j\Omega}$. Then the frequency response of the IIR filter becomes

$$H(z) = \frac{\sum_{k=0}^m b_k e^{-jk\Omega}}{1 + \sum_{k=1}^n a_k e^{-jk\Omega}} \quad (3)$$

$$\text{or, } H(\Omega) = \frac{Y(\Omega)}{X(\Omega)} = \frac{b_0 + b_1 e^{-j\Omega} + \dots + b_m e^{-jm\Omega}}{1 + a_1 e^{-j\Omega} + \dots + a_n e^{-jn\Omega}} \quad (4)$$

where $\Omega=2\pi(ff_s)$ is the digital frequency, f is the analog frequency, and f_s is the sampling frequency. The commonly used approach to IIR filter design is to represent the problem as an optimization problem with the mean square error (MSE) as the error fitness function, $J(\omega)$ expressed as in (5) [4].

$$J(\omega) = \frac{1}{N_s} [(d(p) - y(p))^2] \quad (5)$$

where N_s is the number of samples used for the computation of the error fitness function; $d(p)$ and $y(p)$ are the filter's desired and actual responses. The difference $e(p)=d(p)-y(p)$ is the filter's error signal. The design goal is to minimize the value of error fitness function $J(\omega)$ with proper adjustment of coefficient vector ω represented as: $\omega=[a_0 a_1 \dots a_n \ b_0 b_1 \dots b_m]^T$. In this paper, an improved error fitness function given in (6) is adopted in order to achieve higher stop band attenuation and to have more control on the transition width. Using (6), it is found that the proposed filter design approach results in considerable improvement in stop band attenuation over other optimization techniques.

$$J_1(\omega) = \sum_{\omega} \text{abs}[\text{abs}(|H_d(\omega)| - 1) - \delta_p] + \sum_{\omega} [\text{abs}(|H_d(\omega)| - \delta_s)] \quad (6)$$

For the first term of (6), $\omega \in$ pass band including a portion of the transition band and for the second term of (6); $\omega \in$ stop band including the rest portion of the transition band. The portions of the transition band chosen depend on pass band edge and stop band edge frequencies.

The error fitness function given in (6) represents the generalized fitness function to be minimized using the evolutionary algorithms RGA, conventional PSO and the proposed MFFA, individually. Each algorithm tries to minimize this error fitness J_1 and thus optimizes the filter performance. Unlike other error fitness functions as given in [4], J_1 involves summation of all absolute errors for the whole frequency band, and hence, minimization of J_1 yields much higher stop band attenuation and lesser pass band ripples.

3 Evolutionary Techniques Employed

3.1 Firefly Algorithm (FFA)

Evolutionary techniques RGA and PSO are used to make a comparative study of the results obtained with the proposed optimization technique MFFA and the detailed discussions regarding RGA and PSO are available in [27-28].

3.1.a Behaviour of Fireflies

The flashing light of fireflies which is produced by a bioluminescence process constitutes a signaling system among them for attracting mating partners or potential preys. It is interesting to know that there are about two thousand species of fireflies around the world. Each has its own pattern of flashing. As we know, the light intensity at a particular distance r from the light source obeys the inverse square law. That is to say, the light intensity I decreases as the distance r increases in terms of $I \propto \frac{1}{r^2}$. Furthermore, the air absorbs light. These two combined factors make most fireflies visual to a limit distance.

FFA, developed by Yang [29], is inspired by the flash pattern and characteristics of fireflies. The basic rules for FFA are:

- i. All fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex;
- ii. Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright one will move towards the brighter one, and the brightness decreases as their distance increases. If there is no brighter one than a particular firefly, it will move randomly;
- iii. In this work, the brightness of a firefly is affected or determined by the landscape of the cost function. For a minimization problem, the brightness can simply be inversely proportional to the value of the cost function. The cost function is the error fitness function J_1 in this work.

3.1.b Light Intensity and Attractiveness

The variation of light intensity and formulation of the attractiveness are two important issues in the firefly algorithm. The attractiveness β is proportional; it should be seen in the eyes of the beholder or judged by the other fireflies. Thus it will vary with the distance r_{ij} between firefly i and firefly j . In addition, light intensity decreases with the distance from its source, and light is also absorbed in the media, so the attractiveness will also vary with the degree of absorption. The combined effect of both inverse square law and absorption can be approximated as the following Gaussian form as (7). Hence, the attractiveness function $\beta(r)$ can be any monotonically decreasing function such as the following generalized form:

$$\beta(r) = \beta_0 e^{-\gamma r^m} \quad (m \geq 1) \quad (7)$$

where r is the distance between two fireflies, β_0 is the attractiveness at $r=0$, and γ is a fixed light absorption coefficient which can be used as a typical initial value. In theory, $\gamma \in [0, \infty]$ but in practice γ is determined by the characteristic length of the system to be optimized. In most applications it typically varies from 0.1 to 1. Characteristic distance Γ is the distance over which the attractiveness changes significantly. For a given length scale, the parameter γ can be chosen according to:

$$\gamma = \frac{1}{\Gamma^m} \quad (8)$$

The distance between any two fireflies i and j at x_i and x_j , respectively, is the Euclidean distance as follows:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (9)$$

where $x_{i,k}$ is the k th component of the i th firefly (x_i). The movement of a i th firefly that is attracted to another more attractive (brighter) j th firefly j , is determined by the following equation which shows the relation between the new position of the i th firefly (x'_i) and its old position (x_i):

$$x'_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha \mathcal{E}_i \quad (10)$$

where the second term is due to the attraction. The third term is randomization, with $\alpha \in [0,1]$ being the randomization parameter, and \mathcal{E}_i a vector of numbers drawn from a Gaussian distribution or uniform distribution.

3.1.c Modified Firefly Algorithm (MFFA)

The basic Firefly algorithm is very efficient. It is suitable for parallel implementation because different fireflies can work almost independently. To improve the solution quality, randomness is reduced so that the algorithm could converge to the optimum solution more quickly. Hence, the randomization parameter α decreases gradually as the optima are approaching.

$$\alpha = \alpha_{\infty} + (\alpha_0 - \alpha_{\infty})e^{-t} \tag{11}$$

where $t \in [0, t_{\max}]$ is the pseudo time for simulation and t_{\max} is the maximum number of generations. α_0 is the initial randomization parameter while α_{∞} is the final value. In addition, an extra term $\lambda \epsilon_i (x_i - g_{best})$ is added to the updating formula [26]. In the simple version of the FFA, the current global best g_{best} is only used to decode the final best solutions. The modified updating formula for firefly position is shown in (12).

$$x'_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha \epsilon_i + \lambda \epsilon_i (x_i - g_{best}) \tag{12}$$

3.1.d Steps of Implementation of MFFA to the Design of Low Pass IIR Filters

Steps of MFFA are as follows:

Step 1: Generate initial firefly vectors $x_i = (x_{i1}, \dots, x_{iD})$ ($D = 1, \dots, 18; i=1, \dots, 25$). Set the maximum allowed number of iterations to 500; $\beta_0 = 0.6$; $\gamma = 0.2$; and $\alpha = 0.01$; the population size=25; These values were determined as the best values in a series of thirty preliminary trials.

Step 2: Computation of J_i values of the total population.

Step 3: Computation of the initial population based best solution (g_{best}) vector corresponding to the historical population best and least J_i value.

Step 4: Update firefly positions:

- (a) Compute square root (rsqrt) of Euclidian distance between the first particle vector and the second particle vector as per (9);
- (b) Compute β with the help of β_0 as per (7) and update α as per (11);
- (c) If J_i of second particle is $<$ J_i of first particle, then, update the first particle as per (12) with $+\beta_0$ (case of attraction), otherwise with $-\beta_0$, (case of repulsion);
- (d) Update firefly position as per (12).

Step 5: Repeat Step 2 till maximum iteration cycles.

The values of the parameters used for RGA, PSO and MFFA techniques are given in Table I.

4 Simulation Results and Discussions

4.1 Analysis of Magnitude Response of Low Pass IIR Filters

In this paper digital IIR filters are implemented by MATLAB programs and the best simulation results are reported among thirty independent program runs.

Table 1. Control Parameters of RGA, PSO and MFFA

| Parameters | RGA | PSO SO | MFFA |
|----------------------------|---------------------|-----------|----------------|
| Population size | 120 | 2 5 | 25 |
| Iteration Cycle | 500 | 5 00 | 600 |
| Crossover rate | 1 | - | - |
| Crossover | Two Point Crossover | - | - |
| Mutation rate | 0.01 | - | - |
| Mutation | Gaussian Mutation | - | - |
| Selection | Roulette | - | - |
| Selection Probability | 1/3 | - | - |
| C_1 | - | 2.05 | - |
| C_2 | - | 2.05 | - |
| v_i^{\min} | - | 0.01 | - |
| v_i^{\max} | - | 1.0 | - |
| α, λ, β_0 | - | - | 0.01, 0.2, 0.6 |

In this simulation study, equal numbers of numerator and denominator coefficients are considered for 8th order IIR filter. Hence, 18 coefficients are optimized using each algorithm under consideration, independently and their performances are presented for making a comparative study among the algorithms. The parameters of the low pass filter to be designed are: the sampling frequency $f_s = 1\text{Hz}$; Sampling number is taken as 128; Pass band ripple (δ_p) = 0.001, Stop band ripple (δ_s) = 0.0001, pass band normalized edge frequency (ω_p) = 0.45, stop band normalized edge frequency (ω_s) = 0.50. The control parameters' values of RGA, PSO and MFFA used in this work are given in Table 1. Each algorithm is run for thirty times to get its best solutions. The best results are reported in this paper.

Figure 1 shows the gain plot in dB for the designed low pass 8th order IIR filter. Figure 2 shows the normalized gain plot of the 8th order low pass IIR filter. The best optimized denominator coefficients a_k and numerator coefficients b_k for the designed filter have been calculated using RGA, PSO and MFFA and are given in Table 2. Table 2 also shows that the maximum stop band attenuations achieved for the designed IIR filters using RGA, PSO and MFFA are 27.5145 dB, 30.3635 dB, 37.5474 dB, respectively. Pole-zero plots can be obtained with the filter coefficients reported in Table 2 and in Figure 3, the pole-zero plot obtained for the proposed optimization technique MFFA is only reported. From Figure 3, it is evident that the filter designed using the MFFA is stable as the poles are located within the unit circle.

Table 2. Optimized Coefficients and Performance Comparison of Different Algorithms

| Algorithms | Numerator Coefficient (b_k) | Den. Coeff. (a_k) | Max. stop Band Attenuation (dB) |
|------------|---------------------------------|-----------------------|---------------------------------|
| RGA | 0.0415 0.1234 | 0.9994 -1.1555 | 27.5145 |
| | 0.2676 0.3806 | 2.7421-2.3022 | |
| | 0.4206 0.3484 | 2.4552 -1.4037 | |
| | 0.2164 0.0925 | 0.7776 -0.2480 | |
| | 0.0233 | 0.0524 | |
| PSO | 0.0413 0.1241 | 1.0001 -1.1546 | 30.3635 |
| | 0.2668 0.3791 | 2.7413 -2.3016 | |
| | 0.4202 0.3478 | 2.4547 -1.4044 | |
| | 0.2165 0.0936 | 0.7781 -0.2483 | |
| | 0.0235 | 0.0519 | |
| MFFA | 0.0303 0.0784 | 1.0002 -1.6893 | 37.5474 |
| | 0.1688 0.2386 | 3.3759 -3.4280 | |
| | 0.2710 0.2328 | 3.2821 -2.0259 | |
| | 0.1589 0.0730 | 1.0293 -0.3239 | |
| | 0.0256 | 0.0598 | |

Table 3. Qualitatively Analyzed Data for the 8th Order IIR LP Filter

| Algo-rithm | Maximum Pass band ripple (normalized) | Stop band ripple (normalized) | | | Transition Width |
|------------|---------------------------------------|-------------------------------|------------------------------|---------------------------|------------------|
| | | Maximum ($\times 10^{-2}$) | Minimum ($\times 10^{-4}$) | Mean ($\times 10^{-2}$) | |
| R | 0.0095 | 4.2100 | 15.71 | 2.18 | 0.029 |
| GA | | | 30 | 36 | 7 |
| PS | 0.0021 | 3.0300 | 6.281 | 1.54 | 0.033 |
| O | | | 1 | 64 | 8 |
| M | 0.0024 | 1.2400 | 3.956 | 0.63 | 0.031 |
| FFA | | | 3 | 98 | 9 |

The locations of the zeros, as shown in Figure 3, are positioned outside the unit circle, which implicitly states the system as a non-minimum phase system. Qualitatively analyzed data obtained from Figures 1-2 are reported in Table 3 for all concerned optimization techniques. From Figures 1-2, it is evident that the proposed MFFA based IIR filter design approach produces the highest stop band attenuation and the smallest stop band ripple compared to other optimization techniques. It is also to be noted from Table 3 that the filter designed by the MFFA technique yields quite small transition width, which implies the moderately fast change over from pass band to stop band. The aforementioned statements can be verified from the results presented in Table 3. For both the stop band and pass band regions, the filter designed by the MFFA method results in the improved response than the others.

4.2 Comparative Effectiveness and Convergence Profiles of RGA, PSO and MFFA

In order to compare the algorithms in terms of the error fitness values, Figures 4-5 show the plots of error fitness values against the number of iteration cycles when RGA, PSO and MFFA are employed, respectively, for the design of 8th order IIR low pass filter. From the aforementioned figures, it is seen that the MFFA technique takes 562 iteration cycles to attain the error value of 1.925; whereas, 361 and 359 iteration cycles are required to achieve error values of 2.85 and 4.054 for PSO and RGA techniques, respectively. With a view to the above fact, it may finally be inferred that the performance of the MFFA technique is better as compared to RGA and PSO in terms of the lowest error fitness value in designing the optimal IIR filter. All optimization programs were run in MATLAB 7.5 version on core (TM) 2 duo processor, 3.00 GHz with 2 GB RAM.

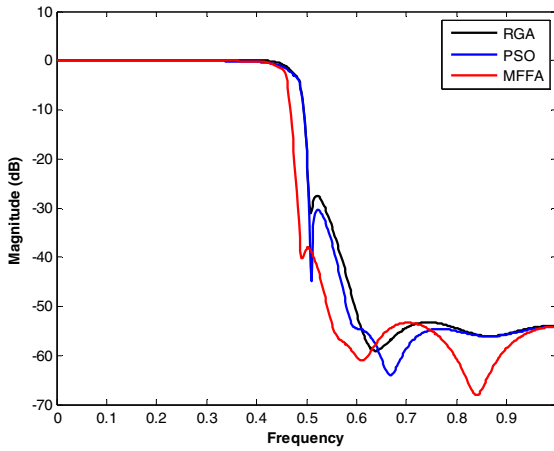


Fig. 1. Gain Plot in dB for the 8th order low pass IIR Filter

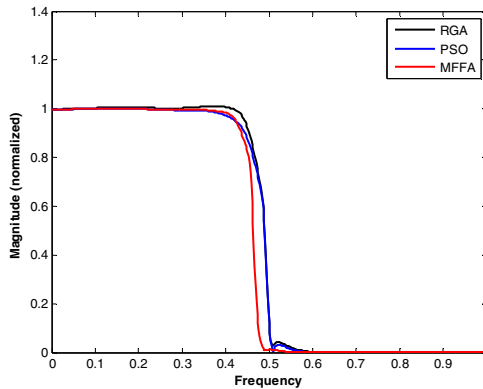


Fig. 2. Normalized gain plot for the 8th order low pass IIR Filter

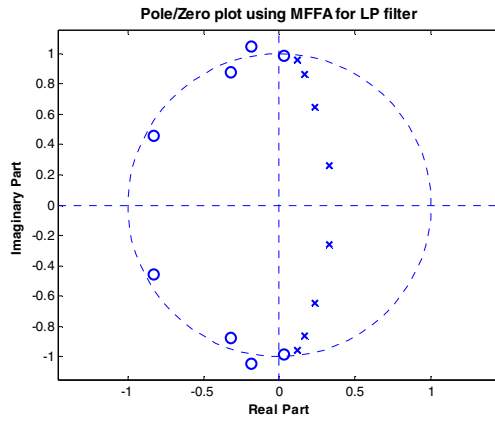


Fig. 3. Pole-Zero plot for the 8th order low pass IIR filter using the MFFA

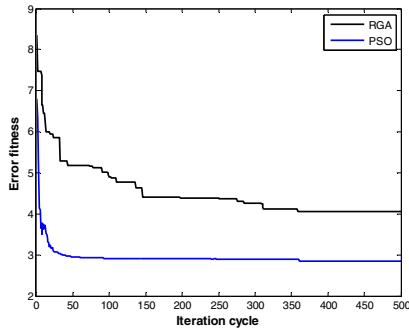


Fig. 4. Convergence profiles for RGA and PSO for the 8th order low pass IIR filter

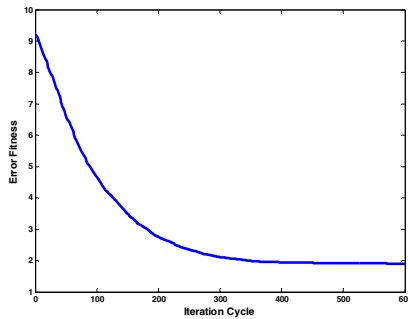


Fig. 5. Convergence profile for the MFFA for the 8th order low pass IIR filter

5 Conclusions

In this paper a recently proposed heuristic search algorithm MFFA is used for the design of IIR LP filter. The modifications adopted in random parameter and position updating process implemented in the MFFA result in better exploration and exploitation of the search space along with the convergence to near-optimal solution. A comparative study between the proposed technique and other well accepted algorithms RGA and PSO affirms that the proposed MFFA based design technique not only provides the highest stop band attenuation but also the quality output in terms of ripples and transition width, which are much better than others. Also the proposed technique attains the lowest value of error fitness function within minimum number of iteration cycles and hence the MFFA is adequate enough for handling other related filter design problems.

References

1. Oppenheim, A.V., Schafer, R.W., Buck, J.R.: *Discrete-Time Signal Processing*. Prentice-Hall, NJ (1999)
2. Hussain, Z.M., Sadik, A.Z., O'Shea, P.: *Digital Signal Processing- An Introduction with MATLAB Applications*. Springer (2011)
3. Xue, L., Rongchun, Z., Qing, W.: Optimizing the design of IIR filter via genetic algorithm. In: *Proc. IEEE Int. Conf. on Neural Networks and Signal Processing*, vol. 1, pp. 476–479 (2003)
4. Karaboga, N., Cetinkaya, B.: Design of minimum phase digital IIR filters by using genetic algorithm. In: *Proc. IEEE 6th Nordic Signal Processing Symposium, Finland*, pp. 29–32 (2004)
5. Dai, C., Chen, W., Zhu, Y.: Seeker optimization algorithm for digital IIR filter design. *IEEE Trans. on Industrial Electronics* 57(5), 1710–1718 (2010)
6. Ahmad, S.U., Andreas, A.: Cascade-form multiplier less FIR filter design using orthogonal genetic algorithm. In: *IEEE Int. Symp. on Signal Processing and Info. Tech.*, pp. 932–937 (2006)
7. Tsai, J.T., Chou, J.H., Liu, T.K.: Optimal design of digital IIR filters by using hybrid Taguchi genetic algorithm. *IEEE Trans. on Industrial Electronics* 53(3), 867–879 (2006)
8. Karaboga, D., Horrocks, D.H., Karaboga, N., Kalinli, A.: Designing digital FIR filters using Tabu search algorithm. In: *IEEE Int. Symp. on Circuits and Systems*, vol. 4, pp. 2236–2239 (1997)
9. Chen, S.: IIR Model Identification using Batch-Recursive Adaptive Simulated Annealing Algorithm. In: *Proc. 6th Annual Chinese Auto. and Comp. Sc. Conf.*, pp. 151–155 (2000)
10. Karaboga, N.: A New Design Method Based on Artificial Bee Colony Algorithm for Digital IIR Filters. *Journal of the Franklin Institute* 346(4), 328–348 (2009)
11. Karaboga, N., Cetinkaya, B.: Design of Digital FIR Filters using Differential Evolution Algorithm. *Circuits Systems Signal Processing* 25(5), 649–660 (2006)
12. Panda, G., Pradhan, P.M., Majhi, B.: IIR System Identification Using Cat Swarm Optimization. *Expert Systems with Applications* 38(10), 12671–12683 (2011)
13. Kalinli, A., Karaboga, N.: Artificial Immune Algorithm for IIR Filter Design. *Engineering Applications of Artificial Intelligence* 18(8), 919–929 (2005)

14. Najjarzadeh, M., Ayatollahi, A.: FIR Digital Filters Design: Particle Swarm Optimization Utilizing LMS and Minimax Strategies. In: IEEE Int. Symp. on Signal Processing and Information Technology, pp. 129–132 (2008)
15. Krusienski, D.J., Jenkins, W.K.: A Modified Particle Swarm Optimization Algorithm for Adaptive Filtering. In: IEEE Int. Symp. on Circuits and Systems, pp. 137–140 (2006)
16. Das, S., Konar, A.: A Swarm Intelligence Approach to the Synthesis of Two-Dimensional IIR Filters. *Engineering Applications of Artificial Intelligence* 20(8), 1086–1096 (2007)
17. Saha, S.K., Kar, R., Mandal, D., Ghoshal, S.P.: Gravitation Search Algorithm: Application to the Optimal IIR Filter Design. *Journal of King Saud University - Engineering Sciences* (2012), doi:<http://dx.doi.org/10.1016/j.jksues.2012.12.003>
18. Saha, S.K., Mukherjee, S., Mandal, D., Kar, R., Ghoshal, S.P.: Gravitational search algorithm in digital FIR low pass filter design. In: Third IEEE Int. Conf. on Emerging Applications of Information Technology (EAIT), pp. 52–55 (2012)
19. Saha, S.K., Kar, R., Mandal, D., Ghoshal, S.P., Mukherjee, V.: A New Design Method Using Opposition-Based BAT Algorithm for IIR System Identification Problem. *Int. J. Bio-Inspired Computation* 5(2), 99–132 (2013)
20. Yang, X.S.: Multi-objective firefly algorithm for continuous optimization. *Engineering with Computers* 29(2), 175–184 (2013)
21. Gandomi, A.H., Yang, X.S., Talatahari, S., Alavi, A.H.: Firefly algorithms with chaos. *Communications in Nonlinear Science and Numerical Simulation* 18(1), 89–98 (2013)
22. Fister, I., Fister Jr., I., Yang, X.S., Brest, J.: A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation* (2013), <http://dx.doi.org/10.1016/j.swevo.2013.6.1>
23. Gandomi, A.H., Yang, X.S., Alavi, A.H.: Mixed variable structural optimization using firefly algorithm. *Computers and Structures* 89(23-24), 2325–2336 (2011)
24. Yang, X.S.: Review of meta-heuristics and generalised evolutionary walk algorithm. *International Journal of Bio-inspired Computation* 3(2), 77–84 (2011)
25. Yang, X.S.: Firefly algorithm, stochastic test functions and design optimization. *International Journal of Bio-inspired Computation* 2(2), 78–84 (2010)
26. Shafaati, M., Mojallali, H.: Modified Firefly Optimization for IIR System Identification. *Control Engineering and Applied Informatics* 14(4), 59–69 (2012)
27. Saha, S.K., Kar, R., Mandal, D., Ghoshal, S.P.: IIR Filter Design with Crazyness Based Particle Swarm Optimization Technique. *World Academy of Science, Engineering and Technology* 60, 1628–1635 (2011)
28. Saha, S.K., Kar, R., Mandal, D., Ghoshal, S.P.: Optimal IIR filter Design Using Novel Particle Swarm Optimization Technique. *Int. Journal of Circuits, Systems and Signal Processing* 6(2), 151–162 (2012)
29. Yang, X.-S.: Firefly Algorithms for Multimodal Optimization. In: Watanabe, O., Zeugmann, T. (eds.) SAGA 2009. LNCS, vol. 5792, pp. 169–178. Springer, Heidelberg (2009)

Firefly Algorithm with Various Randomization Parameters: An Analysis

Nadaradjane Sri Madhava Raja¹, K. Suresh Manic², and V. Rajinikanth³

^{1,3}Department of Electronics and Instrumentation, St. Joseph's College of Engineering, Chennai 600 119, Tamilnadu, India

²School of Engineering, Taylor's University, Lakeside Campus, Malaysia

Abstract. In recent years, metaheuristic algorithms are widely employed to provide optimal solutions for engineering optimization problems. In this work, a recent metaheuristic Firefly Algorithm (FA) is adopted to find optimal solution for a class of global benchmark problems and a PID controller design problem. Until now, few research works have been commenced with FA. The updated position in a firefly algorithm mainly depends on parameters such as attraction between fireflies due to luminance and randomization operator. In this paper, FA is analyzed with various randomization search strategies such as Lévy Flight (LF) and Brownian Distribution (BD). The proposed method is also compared with the other randomization operator existing in the literature. The performance assessment between LF and BD based FA are carried using prevailing parameters such as search time and accuracy in optimal parameters. The result evident that BD based FA provides better optimization accuracy, whereas LF based FA provides faster convergence.

Keywords: Firefly Algorithm, Lévy Flight, Brownian Distribution, Global benchmark problem, Performance measure.

1 Introduction

In the literature, there exist a number of traditional unconstrained optimization methods such as steepest decent method, Newton's method and Quasi-Newton's method to find solutions for a class of linear and nonlinear problems [1]. Traditional methods sometime fail to provide appropriate solutions for complex nonlinear and non-differential problems. Hence, a number of nature inspired stochastic algorithms are proposed by scientists to reduce the complexity of optimization problems. Particularly in 21st century, the scientists proposed a considerable number of heuristic and metaheuristic algorithms such as Bacterial foraging optimization (BFO) [2], Artificial bee colony optimization [3], Glowworm swarm optimization [4], Bat algorithm [5], Firefly algorithm [6], and Cuckoo search [7] to find optimal solutions for more complex and difficult real life optimization problems.

All the metaheuristic algorithms employ firm tradeoff in randomization and local search operation to provide better solution. The selection of a particular algorithm for an optimization problem mainly depends on the following constraints;

(i) Dimensions of search space, (ii) exploration time, (iii) convergence rate, (iv) optimization accuracy and (v) efficiency. Further, number of initial parameters of the algorithm also plays a major role in selecting an algorithm for optimization.

In this work, the FA, initially proposed by Yang is adopted [7]. This algorithm is developed by mimicking the social performance of fireflies. Gandomi *et. al* reported that, the classical FA is conceptually similar to the well known BFO algorithm [8]. In FA, the attraction between two fireflies depends on the distance and the cost function (luminance). From the recent literature, it is observed that, the firefly algorithm is adopted by most of the researchers to find optimal solution for variety of engineering problems [13, 15-20].

2 Firefly Algorithm

Firefly algorithm is a nature inspired metaheuristic algorithm, developed with the inspiration of flashing illumination patterns generated by invertebrates such as glowworm and firefly. They generate chemically produced light from their lower abdomen. This bioluminescence with varied flashing patterns generated by glowworm/firefly is used to establish communication between two neighboring insects, to search for prey and also to find mates.

The classical FA is developed by considering the following conditions [6 - 12]

- (i) All fireflies are unisex and a firefly gets attracted with other nearest one.
- (ii) The attractiveness between two fireflies is proportional to the luminance. In a region, if all fireflies have lesser luminance, they move randomly in a dimensional search space ' D ' until firefly with brighter luminance is found.
- (iii) The brightness of a firefly is somehow related with the analytical form of the cost function assigned to guide the search process.
 - a) For a maximization problem, luminance of a firefly is considered to be proportional to the value of cost function, (i.e., luminance = cost function).
 - b) For a minimization problem, luminance of a firefly is inversely proportional to the value of the cost function, (i.e., luminance = $1/\text{cost function}$).

The overall performance (exploration time, speed of convergence, efficiency and optimization accuracy) of the FA depends on the cost function. In the presented work, for a controller design problem, cost function is chosen as a minimization problem. During the search, without loss of generality, the optimization problem minimizes a scalar function ' J ' of some decision variable vector ' D ' in a universe ' U '. The cost function is framed by assuming, at least there exist one set of optimal parameters in ' U ' which satisfies all the constraints. The minimization problem of constrains can be mathematically expressed as [1];

$$\min_{D \in U} J(D) \quad (1)$$

2.1 Firefly Algorithm Fundamentals

The chief parameters which decide the efficiency of the FA are the variations of light intensity and attractiveness between neighboring fireflies. These two parameters will be affected with increase in the distance between fireflies [6-12].

Variation in luminance of a firefly can be analytically expressed with the following Gaussian form:

$$I(r) = I_0 e^{-\gamma d^2} \tag{2}$$

where I is the light intensity, I_0 is the original light intensity, and γ is the light absorption coefficient.

The attractiveness of firefly towards the luminance can be analytically represented as:

$$\beta = \beta_0 e^{-\gamma d^2} \tag{3}$$

where β is attractiveness coefficient, and β_0 is the attractiveness at $r = 0$.

The above equation describes a characteristic distance $\Gamma = I/\sqrt{\gamma}$ over which the attractiveness changes significantly from β_0 to $\beta_0 e^{-1}$. The attractiveness function $\beta(d)$ can be any monotonically decreasing functions such as the following generalized form;

$$\beta(d) = \beta_0 e^{-\gamma d^m}, (m \geq 1) \tag{4}$$

For a fixed γ , the characteristic length becomes;

$$\Gamma = \gamma^{-1/m} \rightarrow 1, m \rightarrow \infty \tag{5}$$

Conversely, for a given length scale Γ , the parameter γ can be used as a typical initial value (that is $\gamma = 1/\Gamma^m$).

The Cartesian distance between two fireflies i and j at x_i and x_j , in the n dimensional search space can be mathematically expresses as;

$$d_{ij}^t = \|X_j^t - X_i^t\|_2 = \sqrt{\sum_{k=1}^n (X_{j,k} - X_{i,k})^2} \tag{6}$$

In FA, the light intensity at a particular distance d from the light source X_i^t obeys the inverse square law. The light intensity of a firefly I , as the distance d increases in terms of $I \propto 1/d^2$. The movement of the attracted firefly i towards a brighter firefly j can be determined by the following position update equation;

$$X_i^{t+1} = X_i^t + \beta_0 e^{-\gamma d_{ij}^2} (X_j^t - X_i^t) + \Psi \tag{7}$$

where, X_i^{t+1} = updated position of firefly, X_i^t = initial position of firefly,

$\beta_0 e^{-\gamma d_{ij}^2} (X_j^t - X_i^t)$ = attraction between fireflies, and Ψ = randomization parameter.

2.2 Randomization Parameters

The updated position of the i^{th} firefly depends on the values such as initial position of the firefly, attractiveness of firefly towards the luminance, and the randomization parameter (Eqn.7). The convergence speed and optimization accuracy of the firefly mainly depends on the randomization parameter. In the literature, there exist a number of randomization parameters as depicted in Table 1.

Table 1. Various randomization parameters for firefly algorithm

| S. No | Randomization Parameter | Values | Literature |
|-------|--|--|------------|
| 1. | $\alpha \varepsilon_i$ | α = randomization operator ε_i = random variable drawn from a Gaussian distribution | [7, 8, 10] |
| 2. | $\alpha \cdot \varepsilon \cdot Q \text{ rand} ()$ | α = randomization operator $Q \text{ rand} ()$ = random generated quaternion vector | [14] |
| 3. | $\alpha \cdot N_i (0, 1)$ | α = randomization operator N_i = random number drawn from a Gaussian distribution | [14] |
| 4. | $\alpha (2 * \text{rand} - 1)$ | α = randomization operator rand = random number (0,1) | [15] |
| 5. | $\alpha (\text{rand} - 1/2)$ | α = randomization operator rand = random number (0,1) | [16] |
| 6. | $\alpha \cdot \text{sign}(\text{rand} - 1/2) \oplus \text{Lévy}$ | α = randomization operator sign = random sign or direction rand = random number (0,1) | [8, 9] |

2.3 Working Principle

The working principle of a traditional firefly algorithm is demonstrated in this section using a three dimensional optimization problem. The total number of fireflies is assigned as ten. When the algorithm is initialized, all the fireflies are randomly distributed in the search universe as depicted in Fig 1(a). In this optimization problem, it is assumed that, the search space has two local best values and a global best value as shown in Fig 1(b). During the initial search, some firefly move towards the Local Best (LB) values and some reaches the Global Best (GB) value as illustrated in Fig 1 (c). From Fig 1 (c), it is observed that, firefly1 is at LB1; firefly3 is at GB and firefly2 lies between LB1 and GB. The light intensity produced by firefly3 is brighter than the light intensity by firefly1. At this condition, firefly2 moves towards LB1 or GB based on the Cartesian distance ‘d’ (Eqn.7). In this problem, the distance between firefly1 and firefly2 (d1) is less compared to d2, hence firefly2 moves towards LB1 than GB. Similarly, the Cartesian distance between firefly4 and firefly5 (d4) is shorter than d3, and firefly4 is likely attracted towards LB2 than GB.

When the search iteration increases, the firefly at the GB is retained. The attraction signal between the fireflies at the local best value is exponentially decreased with increase in search iteration and it will moves towards the GB. Finally a considerable amount of fireflies are gathered at the global best value as shown in Fig 1(d) at the end of optimization search.

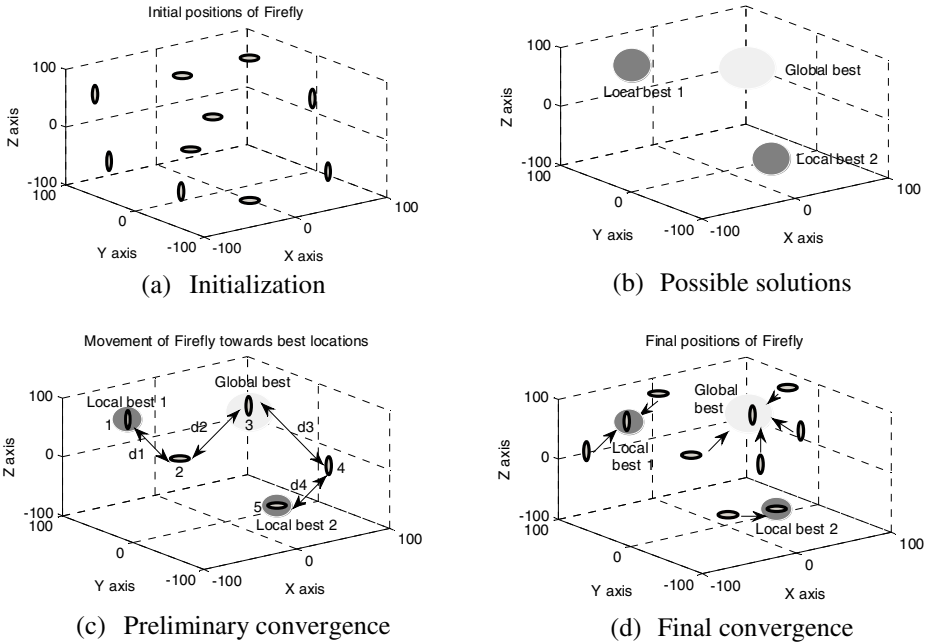


Fig. 1. Various stages in traditional firefly algorithm

3 Lévy Flight and Brownian Distribution

In recently developed nature inspired methods such as firefly and cuckoo algorithm, optimization search process is guided by Lévy Flight (LF) strategy [7].

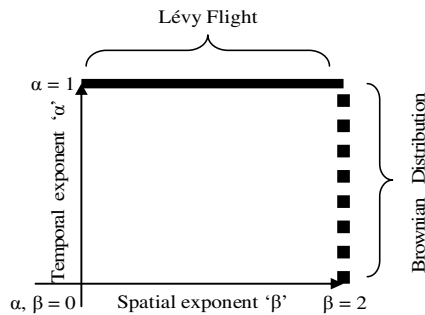


Fig. 2. Relation between LF and BD

The work by Rajasekhar *et. al* also reported that, Lévy guided algorithm performs well compared to the conventional methods [21]. LF is a random walk with a sequence of arbitrary steps and is conceptually similar to the path traced by a molecule as it travels in a liquid or a gas, and the search path of a foraging animal

[21]. In LF, the flight span and the length between two successive changes in direction are drawn from a probability distribution. Like LF, Brownian Distribution (BD) is also in the family of random walks. Fig 2 shows the relationship between LF and BD. Based on the temporal exponent (α) and spatial exponent (β) values; LF and BD can be realized from the random walks [22, 23]. A detailed explanation of LF and BD is provided in the book by Yang [7]. Lévy flight is superdiffusive markovian process, whose step length is drawn from the Lévy distribution in terms of a simple power-law formula;

$$L(s) \sim |s|^{-1-\beta} \text{ where } 0 < \beta \leq 2. \tag{8}$$

The Brownian walk is a subdiffusive non-markovian process, which obeys a Gaussian distribution with zero mean and time-dependent variance. The ratio of the exponents α / β provides the relationship between sub and super diffusion. When $\beta < 2\alpha$ the undecided continuous random walk is successfully superdiffusive, and for $\beta > 2\alpha$ effectively subdiffusive. For $\beta = 2\alpha$, the search process exhibits the same scaling as ordinary Brownian motion [23].

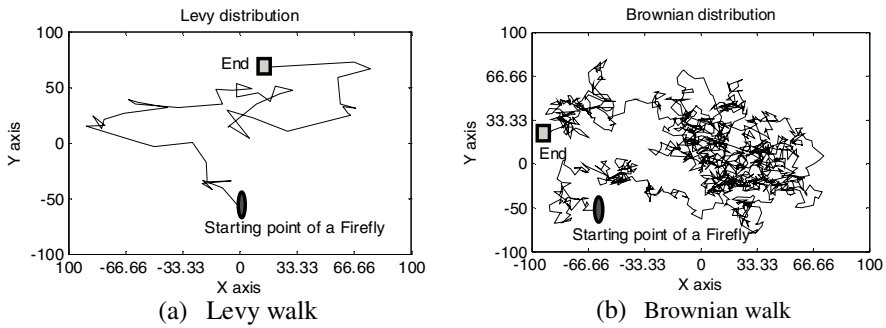


Fig. 3. Search patterns of random walk strategies

Fig 3 depicts the search patterns of firefly algorithm with LF and BD in a two dimensional search space. Fig 3 (a) shows that, Lévy flight guided FA is very efficient in exploring unknown search space with minimal number of iterations, because of its large step size. Fig 3 (b) describes that, the BD guided FA explores the search space with smaller step size and provides the best possible solution.

In this work, the following formulae are considered;

$$L(s) = A \cdot |s|^{1/\beta} \tag{9}$$

$$B(s) = A \cdot |s|^{\alpha/2} \tag{10}$$

$$A = \beta \Gamma(\beta) \sin\left(\frac{\beta\pi}{2}\right) \frac{1}{\pi} \tag{11}$$

where A is the random variable, β is the spatial exponent, α is the temporal exponent, and $\Gamma(\beta)$ is a Gamma function. Eqn. 9 represents the Lévy flight and Eqn. 10

represents Brownian distribution. Both these equations are formed as discussed in [23]. The random variable presented in Eqn. 11 is chosen based on the article by Gandomi *et. al* [8].

4 Results and Discussions

This section presents an evaluation between the traditional FA, Lévy Flight guided FA (LFFA), and Brownian Distribution based FA (BDFa). The proposed and existing FA is initially tested on well known benchmark functions such as Sphere, Rosenbrock, Rastrigin, Griewank and Ackley [3, 21, 24-27]. Later, these algorithms are considered to find the optimal solution for a three dimensional PID controller design problem.

Example 1: Benchmark Functions

This section presents the evaluation of the FA on some well known benchmark functions. Sphere and Rosenbrock are unimodal functions and other functions such as Rastrigin, Griewank, and Ackley are multimodel functions [24-27]. Before initiating the optimization process, it is necessary to assign the values for algorithm parameters based on the guidelines provided by Yang *et. al* [10]. For the global optimization problem, following values are assigned: number of fireflies (n) = 50, $\beta_0 = 1$, $\gamma = 5$, $\alpha_0 = 0.5$ (gradually reduced to 0.01 in steps of 0.001 as iterations proceed), and the total number of run is chosen as 25,000. The position update equations considered in this study is given below;

$$X_i^{t+1} = X_i^t + \beta_0 e^{-\gamma d_{ij}^2} (X_j^t - X_i^t) + \alpha \cdot (rand - 1/2) \quad (12)$$

$$X_i^{t+1} = X_i^t + \beta_0 e^{-\gamma d_{ij}^2} (X_j^t - X_i^t) + \alpha \cdot sign(rand - 1/2) \cdot L(s) \quad (13)$$

$$X_i^{t+1} = X_i^t + \beta_0 e^{-\gamma d_{ij}^2} (X_j^t - X_i^t) + \alpha \cdot sign(rand - 1/2) \cdot B(s) \quad (14)$$

Eqn. 12 represents the traditional FA, Eqn. 13 and 14 represent LFFA and BDFa respectively.

Table 2 provides an appraisal between various firefly algorithms using the fitness value and error of the optimized value. Ten independent runs are performed with the FA and LFFA, and five independent runs are executed with BDFa (because of its slower convergence). Fig 4 shows the amount of error in optimal parameters by FA, LFFA, and BDFa. For Sphere, Rastrigin, and Ackley functions, error by BDFa is small. For Rosenbrock and Griewank functions LFFA provides reduced error compared to FA and BDFa. From this analysis, the observation is, the search time taken by the LFFA is very small compared to classical FA and BDFa. The exploration time taken by the BDFa is very large because of its smaller step size.

Table 2. Performance assessment of FF, LFFF, and BDFF on benchmark problems

| Function | Fitness Value | | | Error | | |
|------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| | FA | LFFA | BDFFA | FA | LFFA | BDFFA |
| Sphere | 4.036e ⁻⁰⁰⁹ | 2.822e ⁻⁰¹⁰ | 3.386e ⁻⁰¹¹ | 5.005e ⁻⁰¹⁰ | 8.332e ⁻⁰¹⁰ | 9.473e ⁻⁰¹² |
| Rosenbrock | 1.174e ⁺⁰⁹ | 1.856e ⁺⁰⁶ | 1.003e ⁺⁰¹ | 1.853e ⁺⁰⁸ | 3.921e ⁺⁰⁵ | 6.014e ⁺⁰¹ |
| Rastrigin | 7.003e ⁺⁰¹ | 5.734e ⁻⁰⁸ | 3.615e ⁻⁰⁵ | 0.992e ⁺⁰⁰ | 2.573e ⁺⁰⁰ | 0.082e ⁺⁰⁰ |
| Griewank | 6.053e ⁻⁰⁰² | 1.646e ⁻⁰¹⁰ | 4.713e ⁻⁰⁰² | 4.737e ⁻⁰⁰² | 3.582e ⁻⁰¹⁰ | 1.024e ⁻⁰⁰¹ |
| Ackley | 7.277e ⁺⁰⁰² | 6.056e ⁺⁰⁰¹ | 6.937e ⁺⁰⁰¹ | 1.946e ⁻⁰³ | 0.835e ⁻⁰¹ | 7.662e ⁻⁰⁴ |

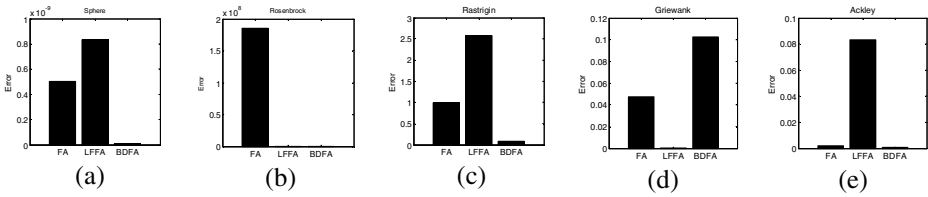


Fig. 4. Graphical representation of error

Example 2: PID Controller Design for Unstable Bioreactor

An unstable bioreactor model discussed by Rajinikanth and Latha is considered [28, 29]. The problem is to design an optimal PID controller, which regulates the feed concentration in order to maintain the product concentration based on the reference input. The unstable first order and second order model of the process is provided in Eqn. 15.

$$G(s) = \frac{-0.9951s - 0.2985}{s^2 + 0.1302s - 0.0509} x e^{-0.1s} = \frac{-5.8644}{5.89s - 1} x e^{-0.1s} \tag{15}$$

For this problem, the firefly algorithm is assigned with the following values: number of fireflies (n) = 10, β₀ = 1, γ = 5, α₀ = 0.5 (gradually reduced to 0.1 in steps of 0.01 as iterations proceed), and the total number of run is chosen as 500. The position update equations (Eqn. 12 - 14) are considered to test the performance of FA, LFFA, and BDFFA. A weighted sum of multiple objective functions is considered in this work as presented in Eqn. 16.

$$J(K_p, K_i, K_d) = w_1 \cdot ISE + w_2 \cdot M_p \tag{16}$$

where, ISE = Error to be minimized, M_p = peak overshoot, and w₁ = w₂ = 1 = weighting function.

The sign of numerator value of the process is negative; hence, the following search boundary is assigned for controller parameters: K_p : min -50% to max 0%; K_i : min -15% to max 0%; and K_d : min -40% to max 0%. The firefly algorithm continuously adjusts the controller parameters until the objective function J is minimised to J_{min} . In this search it is assumed that, the light intensity of the firefly is inversely proportional to the objective function. Five independent runs are executed with the algorithms and the best value among the search is presented in Table 3. Fig 5 shows the traces made by a single firefly on a bounded three dimensional search space for LFFA and BDFA. Fig 6 shows the search path of all the assigned fireflies in Lévy flight assisted FA. From this result, it is observed that, when iteration increases, all the fireflies move towards the optimal value.

Table 3. Optimal value and its performance measure

| Method | K_p | K_i | K_d | Avg. Iteration (5 trials) | ISE | M_p |
|--------|---------|---------|---------|---------------------------|--------|-------|
| FA | -0.7105 | -0.1588 | -0.4026 | 151 | 1.153 | 0.392 |
| LFFA | -0.8032 | -0.1471 | -0.3713 | 94 | 1.344 | 0.340 |
| BDFA | -0.8248 | -0.2116 | -0.4205 | 316 | 0.6494 | 0.361 |

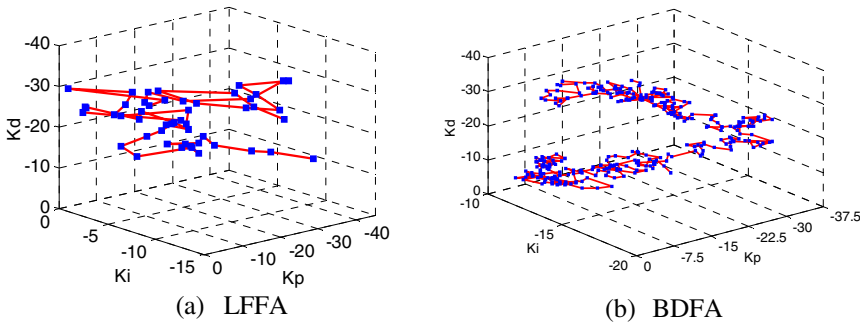


Fig. 5. Traces made by a single firefly in 3D search space

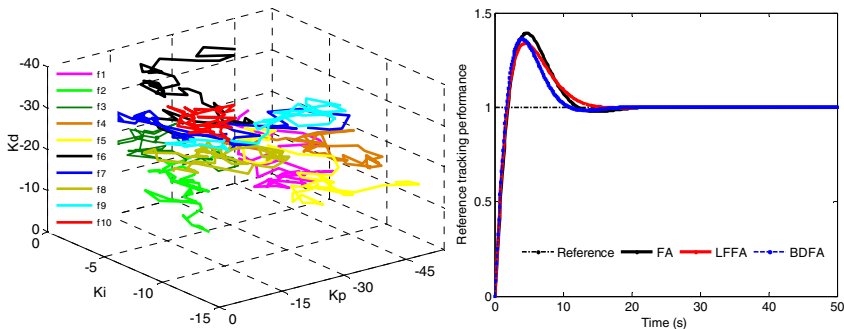


Fig. 6. Optimal PID value attained with LFFA (traces of 10 fireflies)

Fig. 7. Reference tracking performance of bioreactor

Fig 7 depicts the reference tracking performance. From this figure and Table 3, the observation is that, the existing FA requires smaller search time and faster convergence compared to the proposed BDFFA. The LFFFA provides superior performance in search time and convergence rate. The overshoot is also smaller compared to FA and BDFFA. Even though the convergence rate and search time is large, the BDFFA offers enhanced optimization accuracy in rise time, ISE, and settling time compared to FA and LFFFA.

5 Conclusion

In this paper, an analysis on firefly algorithm is presented with random walk search strategies such as Lévy Flight (LF) and Brownian Distribution (BD). The proposed method is also appraised with the well known randomization parameters existing in the literature. In order to test the performance of the LF and BD search methods, unconstrained global optimization problems and three dimensional controller design problem are considered. The performance of LF and BD based FA are quantified using the parameters such as search time and accuracy in optimal parameters. The result of the PID controller design problem evident that, the Brownian search assisted firefly algorithm provides better accuracy in controller parameters, whereas Lévy search based firefly algorithm provides improved search time compared to the existing randomization parameter.

References

1. Guo Ping Liu, G., Yang, J.-B., James Ferris Whidborne, J.: *Multiobjective Optimization and Control*. Printice Hall, New Delhi (2008)
2. Kevin, M.: Passino: Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine* 22(3), 52–67 (2002)
3. Basturk, B., Karaboga, D.: An artificial bee colony (abc) algorithm for numeric function optimization. In: *IEEE Swarm Intelligence Symposium*, Indianapolis, Indiana, USA (2006)
4. Krishnanand, K.N., Ghose, D.: Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications. *Multi-agent and Grid Systems* 2(3), 209–222 (2006)
5. Yang, X.S.: Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation* 3(5), 267–274 (2011)
6. Yang, X.-S.: Firefly Algorithms for Multimodal Optimization. In: Watanabe, O., Zeugmann, T. (eds.) *SAGA 2009*. LNCS, vol. 5792, pp. 169–178. Springer, Heidelberg (2009)
7. Yang, X.S.: *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, UK (2008)
8. Gandomi, A.H., Yang, X.-S., Talatahari, S., Alavi, A.H.: Firefly algorithm with chaos. *Commun. Nonlinear Sci. Numer. Simulat.* 18(1), 89–98 (2013)
9. Yang, X.S.: Firefly algorithm, Lévy flights and global optimization. In: *Research and Development in Intelligent Systems XXVI*, pp. 209–218. Springer, London (2010)

10. Yang, X.-S., Hosseinib, S.S.S., Gandomic, A.H.: Firefly Algorithm for solving non-convex economic dispatch problems with valve loading effect. *Applied Soft Computing* 12(3), 1180–1186 (2012)
11. Yang, X.-S.: Review of meta-heuristics and generalised evolutionary walk algorithm. *International Journal of Bio-inspired Computation* 3(2), 77–84 (2011)
12. Yang, X.-S.: Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-inspired Computation* 2(2), 78–84 (2010)
13. Fister, I., et al.: A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation* (2013), <http://dx.doi.org/10.1016/j.swevo.2013.06.001>
14. Fister, I., Yang, X.-S., Brest, J., Fister Jr., I.: Modified firefly algorithm using quaternion representation. *Expert Systems with Applications* 40(18), 7220–7230 (2013)
15. Poursalehi, N., Zolfaghari, A., Minuchehr, A., Moghaddam, H.K.: Continuous firefly algorithm applied to PWR core pattern enhancement. *Nuclear Engineering and Design* 258, 107–115 (2013)
16. Coelho, L.S., Mariani, V.C.: Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning. *Computers and Mathematics with Applications* 64(8), 2371–2382 (2012)
17. Hassanzadeh, T., Vojodi, H., Mahmoudi, F.: Non-linear Grayscale Image Enhancement Based on Firefly Algorithm. In: Panigrahi, B.K., Suganthan, P.N., Das, S., Satapathy, S.C. (eds.) SEMCCO 2011, Part II. LNCS, vol. 7077, pp. 174–181. Springer, Heidelberg (2011)
18. Rathinam, A., Phukan, R.: Solution to Economic Load Dispatch Problem Based on FIREFLY Algorithm and Its Comparison with BFO, CBFO-S and CBFO-Hybrid. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Nanda, P.K. (eds.) SEMCCO 2012. LNCS, vol. 7677, pp. 57–65. Springer, Heidelberg (2012)
19. Roeva, O., Slavov, T.: Firefly algorithm tuning of PID controller for glucose concentration control during *E. coli* fed-batch cultivation process. In: Proceedings of Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 455–462 (2012)
20. Roeva, O., Slavov, T.: A New Hybrid GA-FA Tuning of PID Controller for Glucose Concentration Control. In: Fidanova, S. (ed.) Recent Advances in Computational Optimization. SCI, vol. 470, pp. 155–168. Springer, Heidelberg (2013)
21. Rajasekhar, A., Abraham, A., Pant, M.: Levy mutated Artificial Bee Colony algorithm for global optimization. In: IEEE International Conference on Systems, Man, and Cybernetics, SMC 2011, pp. 655–662 (2011), doi:10.1109/ICSMC.2011.6083786
22. Nurzaman, S.G., Matsumoto, Y., Nakamura, Y., Shirai, K., Koizumi, S.: From Lévy to Brownian: A Computational Model Based on Biological Fluctuation. *PLoS ONE* 6(2), 016168 (2011), doi:10.1371/journal.pone.0016168
23. Metzler, R., Klafter, J.: The random walk's guide to anomalous diffusion: a fractional dynamics approach. *Physics Reports* 339(1), 1–77 (2000)
24. Farahani, S.M., Abshouri, A.A., Nasiri, B., Meybodi, M.R.: A Gaussian Firefly Algorithm. *International Journal of Machine Learning and Computing* 1(5), 448–453 (2011)
25. 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)
26. Qu, B.-Y., Suganthan, P.N.: Novel Multimodal Problems and Differential Evolution with Ensemble of Restricted Tournament Selection. In: IEEE Congress on Evolutionary Computation (CEC), pp. 1–7 (2010), doi:10.1109/CEC.2010.5586341

27. Arora, S., Singh, S.: The Firefly Optimization Algorithm: Convergence Analysis and Parameter Selection. *International Journal of Computer Applications* 69(3), 48–52 (2013)
28. Rajinikanth, V., Latha, K.: Bacterial foraging optimization algorithm based PID controller tuning for time delayed unstable system. *The Mediterranean Journal of Measurement and Control* 7(1), 197–203 (2011)
29. Rajinikanth, V., Latha, K.: Modeling, Analysis, and Intelligent Controller Tuning for a Bioreactor: A Simulation Study. *ISRN Chemical Engineering* 2012, Article ID 413657, 15 pages (2012), doi:10.5402/2012/413657

Reducing Power Losses in Power System by Using Self Adaptive Firefly Algorithm

B. Suresh Babu^{1,*} and A. Shunmugalatha²

¹Electrical and Electronics Engineering
Odaiyappa College of Engg & Tech
Theni (Dist) - 625531
India

sbeee77@yahoo.com

²Electrical and Electronics Engineering
Velammal College of Engineering & Technology
Madurai – 625 009, Tamil Nadu
India

Abstract . Economic load dispatch (ELD) is an important operational problem of the power system, aiming to reduce the Power loss. The firefly algorithm (FA), a heuristic numeric optimization algorithm inspired by the behavior of fireflies, appears to be a robust and reliable technique. This paper presents a self adaptive FA for the solution of the ELD problem. The proposed algorithm (PA) is applied to the standard IEEE 30 bus test system and the result are presented to demonstrate its effectiveness.

Keywords: economic load dispatch, firefly algorithm.

Nomenclature

| | | | |
|---------------|-------|-------|--|
| a_i | b_i | c_i | Fuel cost coefficients |
| d_i | e_i | | Coefficients of valve point effects of the generator |
| ELD | | | Economic load dispatch |
| FA | | | Firefly algorithm |
| F_C | | | Net fuel cost |
| I_i | | | Light intensity of the i -th firefly |
| $Iter^{\max}$ | | | Maximum number of iterations for convergence check. |
| nd | | | Number of decision variables |
| nf | | | Number of fireflies in the populations |
| ng | | | Number of generators |

* Corresponding author.

| | |
|-------------------------------------|--|
| PA | Proposed algorithm |
| $p(V, \delta)$ | Real power at bus as a function of voltage magnitude and voltage angle |
| $q(V, \delta)$ | Reactive power at bus as a function of voltage magnitude and voltage angle |
| P_{Gi} and Q_{Gi} | Real and Reactive power generation at i -th bus respectively |
| P_{Di} and Q_{Di} | Real and Reactive power demand at i -th bus respectively |
| P_D | Total load demand |
| P_L | Total Transmission losses |
| P_{Gi}^{\min} and P_{Gi}^{\max} | lower and upper limits of P_{Gi} |
| Q_{Gi}^{\min} and Q_{Gi}^{\max} | lower and upper limits of Q_{Gi} |
| r_{ij} | Cartesian distance between the i -th and j -th firefly |
| SAFA | Self adaptive FA |
| t | iteration counter |
| x_i | i -th firefly |
| $\beta_{i,j}$ | Attractiveness between the i -th and j -th firefly |
| β_o and γ | Maximum attractiveness and light intensity absorption coefficient respectively |

1 Introduction

Present day power systems have the problem of deciding how best to meet the varying power demand that has a daily, weekly and yearly cycle in order to maintain a high degree of economy and reliability. Among the options that are available for an engineer in choosing how to operate the system, economic load dispatch (ELD) is the most significant. ELD is a computational process whereby the total required generation is distributed among the generating units in operation also calculate total line losses subject to load and operational constraints. The objective of proposed algorithm to reduce power losses while satisfying various constraints [1].

Over the years numerous methods with various degrees of near-optimality, efficiency, ability to handle difficult constraints and heuristics, are suggested in the literature for solving the dispatch problems. These problems are traditionally solved using mathematical programming techniques such as lambda iteration method, gradient method, linear programming, dynamic programming method and so on. Many of these methods suffer from natural complexity and converge slowly. However, the classical lambda-iteration method has been in use for a long time. The additional constraints such as line flow limits cannot be included in the lambda iteration approach and the convergence of the iterations is dependent on the initial choice of lambda. In large power systems, this method has oscillatory problems that increase the computation time [1,2].

Apart from the above methods, there is another class of numerical techniques called evolutionary search algorithms such as simulated annealing, genetic algorithms, evolutionary programming, ant colony, artificial bee colony and particle swarm optimization have been applied in solving ELD [3-8]. Having in common processes of natural evolution, these algorithms share many similarities; each maintains a population of solutions that are evolved through random alterations and selection. The differences between these procedures lie in the representation techniques they utilize to encode candidates, the type of alterations they use to create new solutions, and the mechanism they employ for selecting the new parents. The algorithms have yielded satisfactory results across a great variety of power system problems. The main difficulty is their sensitivity to the choice of the parameters, such as temperature in SA, the crossover and mutation probabilities in GA and the inertia weight, acceleration coefficients and velocity limits in PSO.

Recently, firefly algorithm (FA) has been suggested for solving optimization problems [9-13]. It is inspired by the light attenuation over the distance and fireflies' mutual attraction rather than the phenomenon of the fireflies' light flashing. In this approach, each problem solution is represented by a firefly, which tries to move to a greater light source, than its own. It has been applied to a variety of ELD problems [14-18] and found to yield satisfactory results. However, the choice of FA parameters is important in obtaining good convergence and global optimal solution.

A self adaptive FA (SAFA) for obtaining the global best solution has been suggested in this paper. The proposed algorithm (PA) has been tested on the IEEE 30 bus test systems to illustrate the performance.

2 Problem Formulation

The ELD problem is formulated as an optimization problem of minimizing the fuel cost while satisfying several equality and inequality constraints. Usually the network loss is calculated using constant B-loss coefficients, which may lead to sub-optimal solution due to approximations in the computations of these coefficients. More accurate solution can be obtained, if network loss is calculated from load flow. In this paper, Newton-Raphson load flow technique [18] is used to calculate the loss. The constrained optimization problem involving load flow is formulated as follows:

$$\text{Minimize } F_C = \sum_{i=1}^{ng} a_i P_{Gi}^2 + b_i P_{Gi} + c_i + \left| d_i \sin(e_i (P_{Gi}^{\min} - P_{Gi})) \right| \quad (1)$$

Subject to:

Real Power balance Constraints

$$\sum_{i=1}^{ng} P_{Gi} - (P_D + P_L) = 0 \quad (2)$$

Real and Reactive power generation limits

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \quad : \quad i = 1, 2, 3 \dots, ng \quad (3)$$

$$Q_{Gi}^{\min} \leq Q_{Gi} \leq Q_{Gi}^{\max} \quad : \quad i = 1, 2, 3 \dots, ng \quad (4)$$

Load flow equations

$$P_{Gi} - P_{Di} - p(V, \delta) = 0 \quad (5)$$

$$Q_{Gi} - Q_{Di} - q(V, \delta) = 0 \quad (6)$$

3 Self Adaptive Firefly Algorithm

The FA is a Meta heuristic, nature-inspired, optimization algorithm which is based on the social flashing behavior of fireflies, or lighting bugs, in the summer sky in the tropical temperature regions. It was developed by Dr. Xin-She Yang at Cambridge University in 2007, and it is based on the swarm behavior such as fish, insects, or bird schooling in nature. It is similar to other optimization algorithms employing swarm intelligence such as PSO and ABC. But FA is found to have superior performance in many cases [9-13]. FA initially produces a swarm of fireflies located randomly in the search space. The initial distribution is usually produced from a uniform random distribution. The position of each firefly in the search space represents a potential solution of the optimization problem. The dimension of the search space is equal to the number of optimizing parameters in the given problem. The fitness function takes the position of a firefly as input and produces a single numerical output value denoting how good the potential solution is. A fitness value is assigned to each firefly. The FA uses a phenomenon known as bioluminescent communication to model the movement of the fireflies through the search space. The brightness of each firefly depends on the fitness value of that firefly. Each firefly is attracted by the brightness of other fireflies and tries to move towards them. The velocity or the pull a firefly towards another firefly depends on the attractiveness. The attractiveness depends on the relative distance between the fireflies. It can be a function of the brightness of the fireflies as well. A brighter firefly far away may not be as attractive as a less bright firefly that is closer. In each iterative step, FA computes the brightness and the relative attractiveness of each firefly. Depending on these values, the positions of the fireflies are updated. After a sufficient amount of iterations, all fireflies converge to the best possible position on the search space. The number of fireflies in the swarm is known as the population

size, nf . The selection of population size depends on the specific optimization problem. However, typically a population size of 20 to 40 is used for PSO and FA for most applications [9-13]. Each i -th firefly is denoted by a vector x_i as

$$x_i = [x_i^1, x_i^2 \dots, x_i^{nd}] \quad (7)$$

The search space is limited by the following inequality

$$x^k(\min) \leq x^k \leq x^k(\max) : k = 1, 2, \dots, nd \quad (8)$$

Initially, the positions of the fireflies are generated from a uniform distribution using the following equation

$$x_i^k = x^k(\min) + (x^k(\max) - x^k(\min)) \times rand \quad (9)$$

Here, $rand$ is a random number between 0 and 1, taken from a uniform distribution. Eq. (9) generates random values from a uniform distribution within the prescribed range defined by Eq. (8). The initial distribution does not significantly affect the performance of the algorithm. Each time the algorithm is executed, the optimization process starts with a different set of initial points. However, in each case, the algorithm searches for the optimum solution. In case of multiple possible sets of solutions, the algorithm may converge on different solutions each time. But each of those solutions will be valid as they all will satisfy the requirements.

The light intensity of the i -th firefly, I_i is given by

$$I_i = Fitness(x_i) \quad (10)$$

The attractiveness between the i -th and j -th firefly, $\beta_{i,j}$ is given by

$$\beta_{i,j} = \beta_o \exp(-\gamma r_{i,j}^2) \quad (11)$$

Where $r_{i,j}$ is Cartesian distance between i -th and j -th firefly

$$r_{i,j} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{nd} (x_i^k - x_j^k)^2} \quad (12)$$

β_o is a constant taken to be 1. γ is another constant whose value is related to the dynamic range of the solution space. The position of firefly is updated in each iterative step. If the light intensity of j -th firefly is larger than the intensity of the

i -th firefly, then the i -th firefly moves towards the j -th firefly and its motion at t -th iteration is denoted by the following equation:

$$x_i(t) = x_i(t-1) + \beta_{i,j} (x_j(t-1) - x_i(t-1)) + \alpha(\text{rand} - 0.5) \quad (13)$$

α is a constant whose value depends on the dynamic range of the solution space. At each iterative step, the intensity and the attractiveness of each firefly is calculated. The intensity of each firefly is compared with all other fireflies and the positions of the fireflies are updated using (12). After a sufficient number of iterations, all the fireflies converge to the same position in the search space and the global optimum is achieved. In the above narrated FA, each firefly of the swarm explores the problem space taking into account the results obtained by others, still applying its own randomized moves as well. The influence of other solutions is controlled by the value of attractiveness of Eq. (11), which can be adjusted by modifying two parameters β_o and γ . The first parameter describes attractiveness at $r_{i,j} = 0$ i.e. when two fireflies are found at the same point of solution space. In general $\beta_o \in [0,1]$ should be used and two limiting cases can be defined: The algorithm performs cooperative local search with the brightest firefly strongly determining other fireflies positions, especially in its neighborhood, when $\beta_o = 1$ and only non-cooperative distributed random search with $\beta_o = 0$. On the other hand, the value of γ determines the variation of attractiveness with increasing distance from communicated firefly. Setting γ as 0 corresponds to no variation or constant attractiveness and conversely putting γ as ∞ results in attractiveness being close to zero which again is equivalent to the complete random search. In general γ in the range of $[0,10]$ can be chosen for better performance. Indeed, the choice of these parameters affects the final solution and the convergence of the algorithm.

The self-adaptive control of these two parameters during the search process effectively leads the algorithm to land at the global best solution with minimum computational effort. Each firefly with nd decision variables in the FA will be defined to encompass $nd + 2$ FA variables in a self-adaptive method, where the last two variables represent β_o and γ . A firefly can be represented as

$$x_i = [x_i^1, x_i^2, \dots, x_i^{nd}, \beta_{o,i}, \gamma_i] \quad (14)$$

Each firefly possessing the solution vector, $\beta_{o,i}$ and γ_i undergo the whole search process of the FA, thereby resulting in better off-springs during the search with lower computational effort. Eq. (11) is accordingly modified as

$$\beta_{i,j} = \beta_{o,i} \exp(-\gamma_i r_{i,j}^2) \quad (15)$$

The self adaptive scheme attempts to prevent sub-optimal solution and enhance the convergence of the algorithm.

4 Proposed Algorithm

The proposed SAFA based solution process involves representation of problem variables β_o and γ ; and formation of a light intensity function.

4.1 Representation of Decision Variables

The decision variables in the PA are real power generation at generator buses except slack bus, β_o and γ . Each firefly in the PA is defined to denote these decision variables in vector form as

$$x = [P_{G2}, \dots, P_{Gng}, \beta_o, \gamma] \quad (16)$$

4.2 Intensity Function

The SAFA searches for optimal solution by maximizing an intensity function, denoted by I_i , which is formulated from the objective function, Eq. (1).

$$\text{Max } I_i = \frac{I}{I + F_C} \quad (17)$$

It is to be noted that the real power generation, which includes network loss, at slack bus is obtained from the load flow.

4.3 Stopping Criterion

The process of generating new swarm can be terminated either after a fixed number of iterations or if there is no further significant improvement in the global best solution.

4.4 Solution Process

An initial swarm of fireflies is obtained by generating random values within their respective limits to every individual in the swarm through Eq. (3). The intensity is calculated by considering the values of each firefly and the movements of fireflies are performed for all the fireflies with a view of maximizing the intensity. The iterative process is continued till convergence. The pseudo code of the PA is as follows.

```

Read the Power System Data
Choose the number of fireflies in the population,  $nf$  and
 $Iter^{\max}$  for convergence check.
Generate the initial population of fireflies
Set the iteration counter  $t=0$ 
while (termination requirements are not met) do
  for  $i=1:nf$ 
    Alter the system data and  $\beta_o \gamma$  according to  $i$ -
    th firefly values
    Run load flow and obtain slack bus power
    Evaluate  $F_C$  and  $I_i$  using Eqs. 1 and 17
    respectively
  for  $j=1:nf$ 
    Alter the system data according to  $j$ -th
    firefly values
    Run load flow and obtain slack bus power
    Evaluate  $F_C$  and  $I_j$  using Eqs. 1 and 17
    respectively
  if  $I_i > I_j$ 
    Compute  $r_{ij}$  using Eq. (12)
    Evaluate  $\beta_{ij}$  using Eq. (15)
    Move  $j$ -th firefly towards  $i$ -th firefly through
    Eq. (13)
  end-(if)
end-( $j$ )
end-( $i$ )
Rank the fireflies
end-(while)

```

5 Simulation

The PA is tested on IEEE 30 bus test system, whose data have been taken from Ref. [19]. The fuel cost coefficients, lower and upper limits for real power generations for IEEE 30 bus test system are given in Table 8.1 of the appendix. Programs are developed in Matlab 7.5 and executed on a 2.3 GHz Pentium-IV personal computer. Newton Raphson technique [18] is used to carry out the load flow during the optimization process. The parameters chosen for the PA are given in Table 1.

Table 1. FA parameters

| Parameter | Value |
|---------------|-------|
| nf | 30 |
| $Iter^{\max}$ | 300 |

Table 2. Results of IEEE 30 bus test system

| Control Variables (p.u) | Before Optimization | PA |
|-------------------------|---------------------|----------------|
| P_{G1} | 138.53900 | 51.787450 |
| P_{G2} | 57.56000 | 80.00000 |
| P_{G5} | 24.56000 | 50.00000 |
| P_{G8} | 35.00000 | 35.00000 |
| P_{G11} | 17.93000 | 30.00000 |
| P_{G13} | 16.91000 | 40.00000 |
| β_o | --- | 0.210124 |
| γ | --- | 0.539176 |
| Load Demand(MW) | 283.40000 | 283.40000 |
| Loss(MW) | 7.09900 | 3.38740 |

The optimal solution obtained by the PA for IEEE 30 bus test system are given along with the initial solution before optimization in Tables 5.2 respectively. It is very clear from these tables that the proposed algorithm is able to total line loss is decreased. The resulting loss after optimization IEEE 30 bus test system is decreased.

6 Summary

Indeed the FA is a powerful novel population based method for solving complex optimization problems. The convergence and searching capability can be improved with a view to prevent sub-optimal solution through self-adaptive control of FA parameters. In this paper, SAFA solution technique for ELD problem is developed and tested on IEEE 30 bus test system. The algorithm uses NR load flow technique for computing the slack bus power that includes network loss and is able to offer the global best solution at lower computational burden.

Acknowledgement. The authors gratefully acknowledge the authorities of Odaiyappa College of Engineering and Technology and Velammal College of Engineering and Technology for their continued support, encouragement and the facilities provided to carry out this work.

Appendix

Table 3. Generator Data for IEEE 30 bus test system

| Bus No | a | b | c | d | e | P_{Gi}^{\min} | P_{Gi}^{\max} |
|--------|---------|------|-----|-----|-----|-----------------|-----------------|
| 1 | 0.00375 | 2.00 | 0 | 0 | 0 | 50 | 200 |
| 2 | 0.01750 | 1.75 | 0 | 0 | 0 | 20 | 80 |
| 5 | 0.06250 | 1.00 | 0 | 0 | 0 | 15 | 50 |
| 8 | 0.00834 | 3.25 | 0 | 0 | 0 | 10 | 35 |
| 11 | 0.02500 | 3.00 | 0 | 0 | 0 | 10 | 30 |
| 13 | 0.02500 | 3.00 | 0 | 0 | 0 | 12 | 40 |

References

1. Wood, A.J., Woolenber, B.: Power generation, operation and control. John Willey and Sons, New York (1996)
2. Chowdhury, B.H., Rahman, S.: A review of recent advances in economic dispatch. IEEE Trans. on Power Systems 5(4), 1248–1259 (1990)
3. Panigrahi, C.K., Chattopadhyah, P.K., Chakrabarti, R.N., Basu, N.: Simulated annealing technique for dynamic economic dispatch. Electric Power Components and Systems 34(5), 577–586 (2006)
4. Adhinarayanan, T., Sydulu, M.: Directional search genetic algorithm applications to economic dispatch of thermal units. International Journal for Computational Methods in Engineering Science and Mechanics 9(4), 211–216 (2008)
5. He, D.-K., Wang, F.-l., Mao, Z.-Z.: Hybrid genetic algorithm for economic dispatch with value-point effect. Electric Power Systems Research 78, 626–633 (2008)
6. 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)
7. Subbaraj, P., Rengaraj, R., Salivahanan, S., Senthilkumar, T.R.: Parallel particle swarm optimisation with modified stochastic acceleration factors for solving large scale economic dispatch problem. Electrical Power and Energy Systems 32(9), 1014–1023 (2010)
8. Pereira-Neto, A., Unsihuay, C., Saavedra, O.R.: Efficient evolutionary strategy optimization procedure to solve the nonconvex economic dispatch problem with generator constraints. IEEE Proc. Gener. Transm. Distrib. 152(5), 653–660 (2005)
9. Yang, X.S.: Nature-Inspired Meta-Heuristic Algorithms. Luniver Press, Beckington (2008)
10. Yang, X.-S.: Firefly algorithms for multimodal optimization. In: Watanabe, O., Zeugmann, T. (eds.) SAGA 2009. LNCS, vol. 5792, pp. 169–178. Springer, Heidelberg (2009)
11. Yang, X.-S.: Firefly algorithm, stochastic test function and design optimisation. International Journal of Bio-inspired Computation 2(2), 78–84 (2010)
12. Yang, X.-S.: Review of metaheuristics and generalized evolutionary walk algorithm. International Journal of Bio-inspired Computation 3(2), 77–84 (2011)
13. Yang, X.-S.: Multiobjective firefly algorithm for continuous optimization. Engineering with Computers 18(2), 175–184 (2013)
14. Swarnkar, K.K.: Economic load dispatch problem with reduce power loss using firefly algorithm. Journal of Advanced Computer Science and Technology 1(2), 42–56 (2012)

15. Vinod Kumar, K., Lakshmi Phani, G.: Combined economic emission dispatch-pareto optimal front approach. *International Journal of Computer Applications* 30(12), 16–21 (2011)
16. Sulaiman, M.H., Mustafa, M.W., Zakaria, Z.N., Aliman, O., Abdul Rahim, S.R.: Firefly Algorithm Technique for Solving Economic Dispatch Problem. In: 2012 IEEE International Power Engineering and Optimization Conference (PEOCO 2012), Melaka, Malaysia, June 6-7 (2012)
17. Chandrasekaran, K., Simon, S.P.: Tuned Fuzzy Adapted Firefly Lambda Algorithm for Solving Unit Commitment Problem. *J. Electrical Systems* 8(2), 132–150 (2012)
18. Tinney, W.F., Hart, C.E.: Power flow solution by Newton's method. *IEEE Trans. PAS-86*, 1449–1460 (1967)
19. TestSystems Archive, <http://www.ee.washington.edu/research/pstca/> (accessed December 2012)

A Soft-Computing Based Approach to Economic and Environmental Analysis of an Autonomous Power Delivery System Utilizing Hybrid Solar – Diesel – Electrochemical Generation

Trina Som and Niladri Chakraborty

Department of Power Engineering, Jadavpur University,
Salt Lake City, Kolkata 700098

trinasom@gmail.com, chakraborty_niladri@hotmail.com

Abstract. Concerns toward the continued availability of reliable grid-based power and global warming and depleting oil reserves have made a decentralized power delivery model seeking energy from renewable energy resources an inevitability. Photovoltaic power generating modules (PV), diesel generators (DG), battery energy storage systems (BESS) are emerging generation/storage technologies. The present work depicts the economic analysis and environmental impacts of a decentralized or distributed power delivery system integrated with hybrid distributed energy resources (DERs). The model for decentralized power delivery system has been developed employing a modified form of the differential evolution algorithm implemented within MATLAB® Simulink considering load demand scenario for a locality in India. Optimal power generation has been made using different sets of distributed energy resources, pertaining to cost estimation and respective environmental impact. The results show a cost effective power delivering network for hybrid DG-BESS, but PV-BESS is more beneficial from the environmental perspective.

Keywords: hybrid-Distributed Energy Resources, Modified Differential Evolution.

1 Introduction

The need for energy-efficient electric power sources in remote locations is a driving force for research in hybrid energy systems. Power utilities in many countries round the world are diverting their attention toward more effective and renewable electric power sources [1,2]. Reasons for this interest include the possibilities of taxes or other penalties for emissions of greenhouse gases as well as other pollutants with finite supply of fossil fuels. The use of renewable energy sources in remote locations could help reduce the operating cost through the reduction in fuel consumption, increase system efficiency, and reduce noise and emissions.

DERs share small to medium-sized markets which are expected to increase in the future leading to a much more decentralized power delivery system as depicted in

literature [3,4]. A number of studies [5-8] related to the design and operations of DERs are available in the literature. Microgrid operations of DERs have also been discussed in the literature [9-13]. Optimization methods based on soft computing has been applied to economic load dispatch and load scheduling problems are available in the literature [14-16]. Application of few modified soft computing algorithm has also been made by Mallipedi. et.al. in dealing with complex constraining problems[17, 18]. Though the technical issues of microgrid have been dealt with by several researchers [19-21] and economic issues [24-27] related to microgrids are also addressed by some studies, but economic evaluation of a decentralized power delivery system with different combinations of DERs and consumer load data specific to the Indian scenario have not been studied. Even if the applications of soft computing techniques in microgrid integrated DERs systems have been found in literature [27-28], but economic evaluations of different types of autonomous decentralized power delivery system, with varying combinations of load and generation through soft-computing based optimization techniques are sparse in literature. Again economic evaluation in consideration of environmental affect of a decentralized power delivery system with different combinations of DERs have hardly been studied through soft-computing based optimization technique in the literature. Due to a huge gap of shortage between power supply and demand, an Indian load demand scenario has been considered for modeling an autonomous power delivery framework integrated with DERs and consumers.

The economic part of the model calculates the fuel consumed per kilowatt-hours for DG and battery energy storage system along with the total initial cost for different DERs, and constructional cost for power delivery network.

The environmental part of the model calculates the CO₂, particulate matter (PM), and the NO_x emitted to the atmosphere [1]. Simulations based on an actual system in the remote locality of India were performed for three cases which consist of different combinations of DERs.

2 Physical Problem:

Economic estimation for distributed power system has been analyzed by optimal power generation from different sets of DERs. The power optimization is based on the logic of meeting the required load demands in apportion to the electrical production between the PV [29], DG [30] and battery energy storage system. The power delivery systems refer to three specific test cases as follows

Case I: Autonomous power delivery system consist of hybrid photovoltaic power generating modules (PV) - diesel generators (DG) - battery energy storage system (BESS) as DERs.

Case II: Autonomous power delivery system consists of hybrid diesel generators (DG) - battery energy storage system (BESS) as DERs.

Case III: Autonomous power delivery system consisting of hybrid photovoltaic power generating modules (PV) - diesel generators (DG) as DERs.

The model for simulation in case I, as shown in figure.1, contain all 3 power generating resources, namely photovoltaic module, diesel generator and battery bank. It generally consists of a Diesel-cycle (based on compression ignition system) reciprocating engine prime mover coupled to an electric generator. The diesel engine operates at a relatively high compression ratio and relatively low speed. Diesel generation sets are proven to be cost effective, extremely reliable and widely used technology. They are manufactured in a wide range of sizes, from about 1 kilowatt (kW) upto about 10 MW. Diesel generators for our experiment purposes have been taken for such a capacity that it fulfills the remaining load requirement, and has been used as a last power generating option in consideration to environmental aspect.

The stimulant model for case III, as shown in figure 3, consists of hybrid PV-DG power generating resources. The optimal power operation is performed by the same logic as used for case I and case II. Here the load power required is first compared with the power supplied by PV, and next the less power generated in comparison with the required demand are then delivered by DG.

2.1 Problem Formulation

The economic analysis part of the simulation model involves calculation of the cost of PV with diesel set along with battery system. The equations (1) (2) and (3) represents the total cost function for case I, case II, and case III respectively as shown below;

$$C_{Total} = \left[\sum_{hr=t_8}^{t_{16}} P_{PV}(hr)O_{PV} + \sum_{hr=t_8}^{t_{16}} P_{DG}(hr)O_{DG} + \sum_{hr=t_1}^{t_{24}} P_{bt}(hr)O_{bt} \right] \tag{1}$$

$$C_{Total} = \left[\sum_{hr=t_8}^{t_{16}} P_{DG}(hr)O_{DG} + \sum_{hr=t_1}^{t_{24}} P_{bt}(hr)O_{bt} \right] \tag{2}$$

$$C_{Total} = \left[\sum_{hr=t_8}^{t_{16}} P_{PV}(hr)O_{PV} + \sum_{hr=t_8}^{t_{16}} P_{DG}(hr)O_{DG} \right] \tag{3}$$

The power generated from solar energy (Pso) depends on certain geographical and environmental factors which is expressed [29] as

$$P_{SO} = \left[(Df)(RC_{PV}) \left(\frac{G_S}{S_S} \right) \right] \tag{4}$$

where, *Df* is the derating factor, *RC_{PV}* is the rated capacity of the solar array, *G_s* and *S_s* are global solar radiation incident on the surface of solar array and standard solar radiation for the rated capacity.

Knowing the efficiency from the manufacturer’s specifications for the efficiency of the electric generator, and the load on the generator, the power input (*P_{input}*) to the generator [1] can be calculated as

$$P_L = P_{input} \times \eta \tag{5}$$

where P_L is the load on the diesel generator. Again the P_{input} is based on fuel consumption, *i.e.* the amount of fuel required by the diesel engine to supply the load. The fuel consumption can be mathematically interpreted by manufacturer's data sheet as [1]

$$F_c (\text{gallon}) \times 0.7 = 0.5 \times P_{input} \quad (6)$$

where is the input power to the generator given in kilowatts; 7.1 is the factor that converts pounds (lbs) to gallons, depending on the type of fuel that is used. For different types of generators, the fuel consumption can be obtained from the manufacturer's data sheet. $\eta = 0.5$.

The total output from a battery can be obtained as total voltage supplied in terms of number of battery cells (n) and the voltage per cell [1] as follows:

$$\text{battery_voltage} = n \times \text{voltage_per_cell} \quad (7)$$

where voltage per cell is obtained from the different electro-chemical reactions.

The environmental analysis is performed on the basis of emission of NOx, PM, CO₂. The amount of emission is obtained from total kilowatt-hours per gallon supplied by the diesel generator. The amount of carbon, NOx, PM emitted per kWh is calculated from the equations shown below as (8) and (9).

$$\text{kWh/gallon} = \text{kWh}_{Gen} / F_c \quad (8)$$

$$\text{Total Cost} = \text{cost/gallon} \quad (9)$$

where kWh is the total kilowatt-hours supplied by the diesel generator, and F_c is the total fuel consumed (in gallons). The amounts of carbon emitted per kWhr have been calculated by dividing the total amount of carbon emitted by total electricity generated by the sources. As per recent records [32] the carbon credit is valued at 10€/ton as of april 2013. The conversion factor results in Rs 700/ton. The CO₂ emission is 1 kg per kWhr [33] of power generation, where the kWhr generated per gallon of fuel is calculated as the product of the density of diesel, *i.e.* 0.832kg/liter and its calorific value, *i.e.* 137213.44 kJ. However these costs are subjected to many constraints that need to be considered for economic cost analysis.

2.2 Constraining Function:

The main constraining factor considered for the problem, defines the exact balance of power demand and power generations from DERs and BESS. Equations (10), (11), and (12) represent the main functional constraints for case I, case II and case III respectively.

$$\sum_{hr=1}^{24} P_{Demand} = P_{SO}(hr) + P_{bt}(hr) + P_{DG}(hr) \quad (10)$$

$$\sum_{hr=1}^{24} P_{Demand} = +P_{DG}(hr) + P_{bt}(hr) \quad (11)$$

$$\sum_{hr=1}^{24} P_{Demand} = P_{SO}(hr) + P_{DG}(hr) \quad (12)$$

Equations (13), (14), and (15) present the side constraining functions. The sides constraints describe the power generated from DERs are within its installed capacity. The power generations are always positive.

$$P_{SO} < IC_{SO} \quad (13)$$

$$P_{DG} < IC_{DG} \quad (14)$$

$$P_{bt} < IC_{bt} \quad (15)$$

$$IC_{SO}, IC_{DG}, IC_{bt} > 0 \quad (16)$$

With the objective function and constraining functions known apriori, a modification over conventional DE has been made and applied for economic analysis of these systems.

3 Control Technique

The numerical method used for the present problem of power optimization is based on the heuristic or soft computing logic. These techniques are most suited for real-world applications that are characterized by imprecise, uncertain data and incomplete domain knowledge. With varying load demand having different consumers and different seasons these type of problems sometimes become too complex to model mathematically. Therefore, soft computing methods have been employed to provide better economic solutions through optimal power operation. The proposed algorithm is made by modification over conventional differential algorithm. The conventional differential algorithm (DE) has been described as;

3.1 Differential Evolution

The differential evolution algorithm (DE) is technically simple and highly efficient technique for constrained parameter optimization problems. The population of solution vectors is successively updated through probabilistic search method [34] which involves the following steps, where at first, a population size of x encoded elements is chosen randomly as initial solution vector for individual variables of the concerned problem. Next, three mutually distinct string of initial solutions are randomly drawn from population vector. Mutation is performed over the solution corresponding to the randomly drawn string of solutions on the basis of mutation probability factor. Further, the crossover operation is performed [35] between individual elements of mutated variables depending upon the cross over probability check. Finally the selections for better solution elements are made by calculating the fitness value, through the evaluation of the objective function for both the initial and mutated solutions. The iteration procedure is continued until the termination condition, *i.e.* difference between the best fitness values of consecutive iterations to become negligible in magnitude is reached.

3.2 Modified Differential Evolution (MDE)

In the present problem, an attempt of using a modified DE (MDE) has been made for achieving an improved result. The modification involves each solution generating more than one offspring using different mutation operators by combining information of the best solution in the population and thereby increasing the probability of generating better offspring. A consideration of improved variants of DE [35] which utilizes the concept of neighborhood of each population member, balances the exploration and exploitation abilities of DE without imposing serious additional burden in terms of function evaluations. An increase in the probability of generating better offspring has been made by sampling the feasible region in a better way and reaching the global optimum solution. The mutation operation in MDE includes two mutant operands:

$$v_{i,j} = x_{r3,j} + F_{\alpha} (x_{best,j} - x_{r2,j}) + F_{\beta} (x_{best,j} - x_{r1,j}) \quad (17)$$

where, F_{α} and F_{β} indicate the influence of the best and parent solution respectively in search direction of off-spring. x_{best} is the best individual, while x_i is the i^{th} individual.

In house code has been developed using Matlab programming language. The optimization method was implemented through Matlab Embedded block in the simulation model for all the three cases. The Matlab embedded function block is the control unit of the simulation models developed for three case studies. The flow diagram of MDE has been shown below in Figure 1.

4 System Modeling and Results

The simulation models have been developed on the basis of the three cases mentioned above. The model for case I has been shown in figure 2. For case I, initially the load power required is compared with the power supplied by the pv module. If the generated power is more than the required power then the excess power is used for the charging of battery bank, while if the power generated by the pv module is not enough to feed the load entirely then the pv module supplies as much power within its installed capacity and the remaining demand is supplied by DG or BESS. This is again optimally generated by comparing the power delivering capacities of diesel generator and BESS, i.e. if the battery energy storage system is not able to supply the required load then it provides upto its installed capacity and finally the remaining load requirement is met by the diesel generator.

The stimulant model for case II, as shown in figure.3, consists of hybrid DG-BESS power generating resources. The optimal power operation is performed by the same logic as used for case I. Here the load power required is compared with the power supplied by DG. If the generated power is less than the required demand then the remaining power is delivered by BESS, or it will charge the BESS.

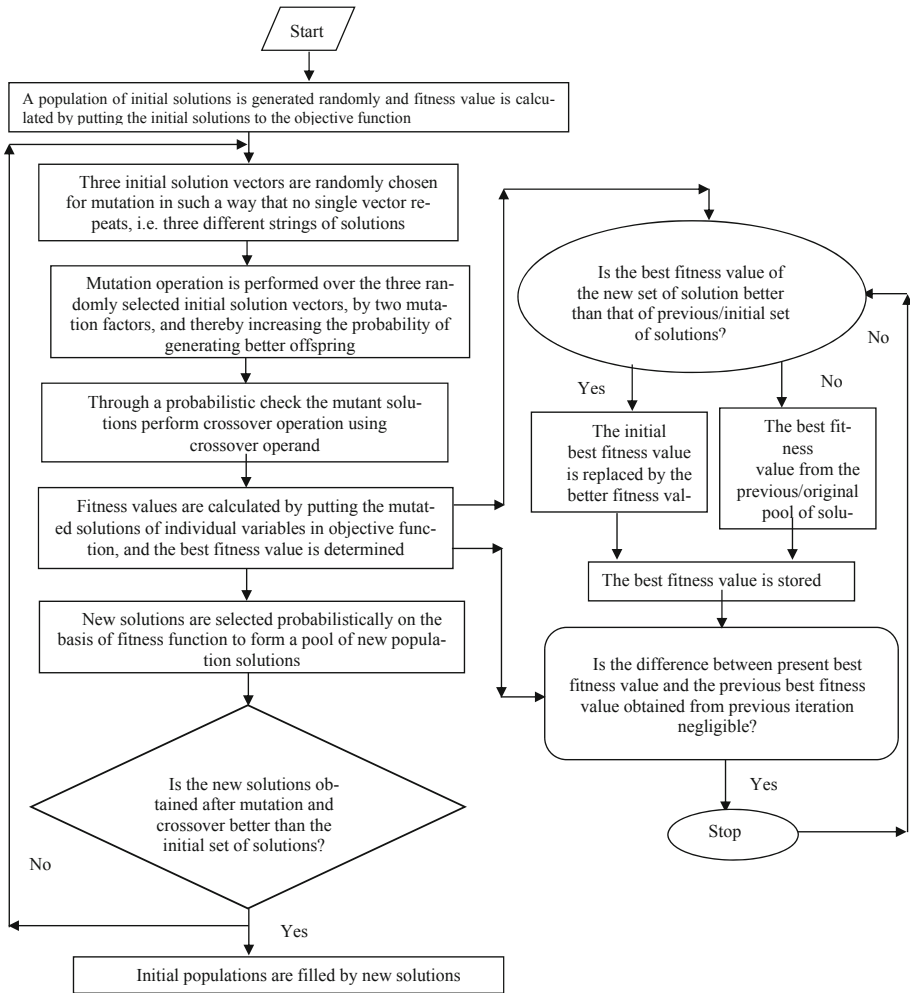


Fig. 1. Flow chart of MDE

Figure.4 represents the stimulant model for case III. This consists of hybrid PV-DG as distributed energy resources. The optimal power operation is performed by the same logic as used for case I and case II. Here the load power required is compared with the power supplied by PV during day time. If the generated power is less than the required demand then the remaining power is delivered by BESS, or if the power generated is more than demand the excess power is fed to BESS for charging, which further helps in delivery power during peak and night time.

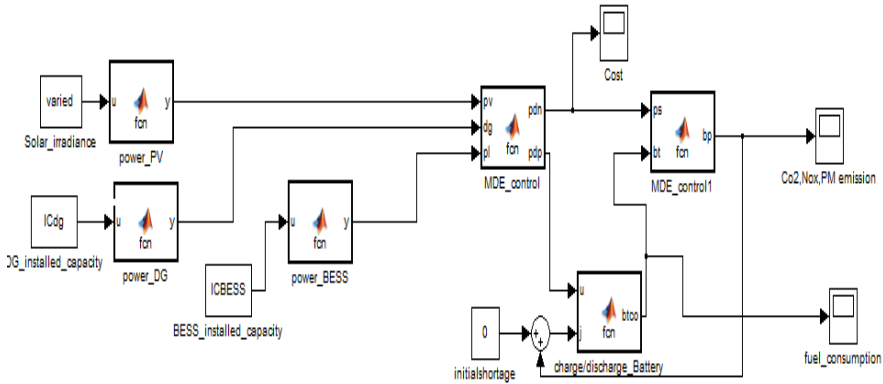


Fig. 2. Simulation Model for case I consisting of hybrid PV-DG-BESS

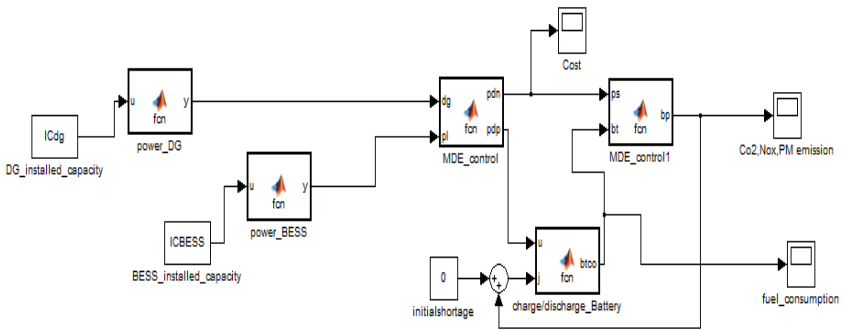


Fig. 3. Simulation Model for case II consisting of hybrid DG-BESS

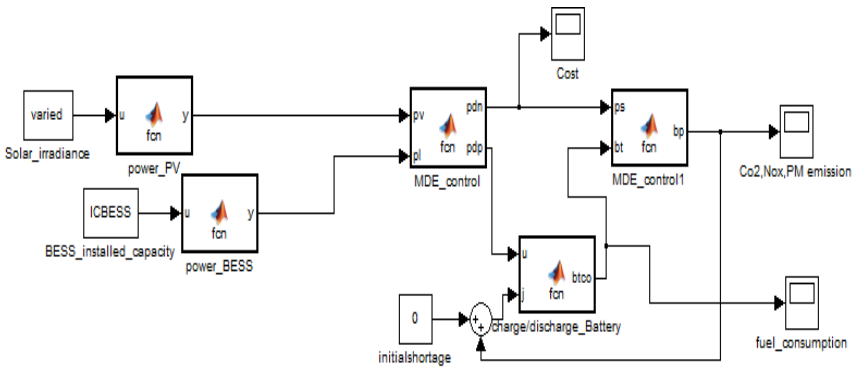


Fig. 4. Simulation Model for case III consisting of hybrid PV-BESS

The input parameters considered for the simulation are load demand data for a real-time Indian scenario, initial costs for different DERs and annual solar irradiance for India [36]. These input parameters are shown in table 1 [37, 38, 39]. A typical

representative of the load demand for a small Indian locality is considered which is obtained by averaging the load data over a year [9].

Table 1. Initial Costs for DERs

| DERs | Ratings | Initial Cost (Rs.) | Running Cost DG (Rs) | Running Cost BESS (Rs) |
|-----------------|-------------------------------------|--------------------|----------------------|------------------------|
| PV, DG and BESS | PV=80 KW DG=50 KW BESS=100 KW | 13432500 | 54 /gallon | 6/kW hr |
| PV and DG | PV=80 KW DG=120 KW | 13640000 | 54 /gallon | 6/kW hr |
| DG and BESS | DG=100KW BESS=120 KW | 787000 | 54 /gallon | 6/kW hr |

The simulated results for case I showing the CO₂emission has been presented in figure 5.

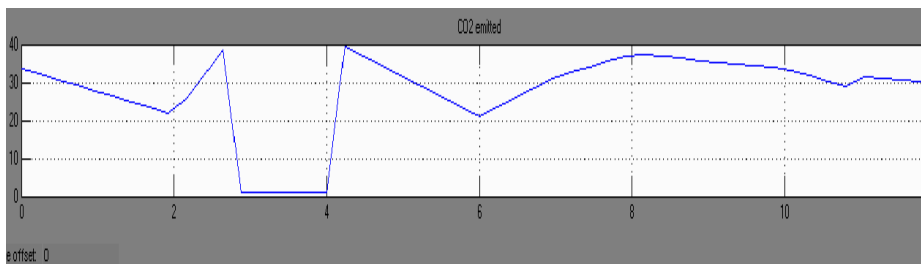


Fig. 5. Emission of CO₂ for case I with respect to simulation time

Similarly, the results for case II and case III in terms of fuel consumption and NO_x emitted have been portrayed in figure 6 and figure 7 respectively.

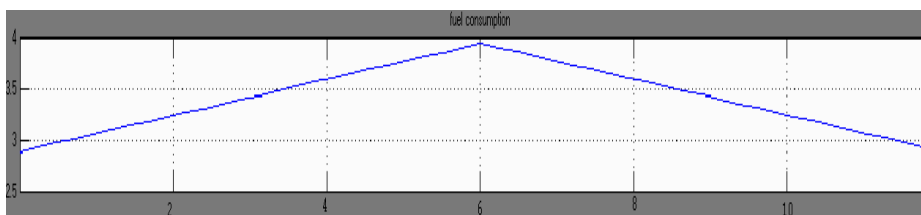


Fig. 6. Fuel consumption for case II with respect to simulation time

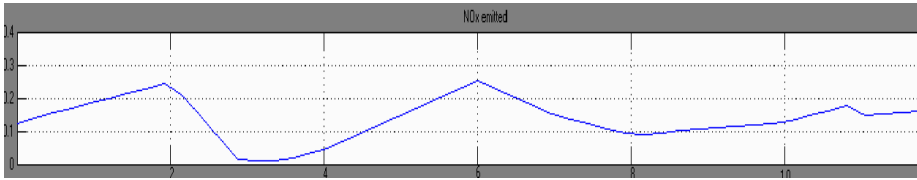


Fig. 7. Emission of NO_x for case III with respect to simulation time

The comparative results between case I, case II and case III has been presented in Table 2 as shown below.

Table 2. Relative Economic and Environmental studies of case I, case II and case III

| DERs outputs | PV-DG- BESS | DG-BESS | PV-DG |
|-------------------------------------|----------------|-------------|-----------|
| Fuel Consumption (Gallons) | 101.2842 | 352.5071 | 140.5393 |
| Total Fuel cost (Rs) | 14969.77 | 52136.1525 | 20785.903 |
| Total Annual Cost(Rs) | 13452575.8 | 13673582.69 | 792852.15 |
| NO _x Emitted (Pounds) | 12.2167 | 42.5476 | 16.9630 |
| CO ₂ Emitted (Pounds) | 1221.663 | 4254.7607 | 1696.3093 |
| PM Emitted (Grams) | 24.165 | 84.1610 | 33.905 |

Though the number of DERs installed are more than that of case II, still the total costs computed by MDE for case II have been found to be more than that of case I. It is mainly due to the running cost, which occurs more in case II for installing more DG capacity for the same power requirement. The fuel cost obtained in case II have been found to be more than that of case I, which is mainly due to the result of optimal power generation from PV as DERs having zero running cost, and thereby with BESS having less fuel consumption. Moreover, though running costs of BESS is less than that of DG, but as the DG set supplies power with PV which is time dependent and can be only used during day time, so the fuel cost for case III is found to be have more fuel cost than that of cost I. Again having the same number of DERs a reduction in fuel cost occurs in case III from that of case II which can be analyzed through optimal power generations of DG with PV in case III.

5 Conclusion

The present work concludes that a decentralized power delivery system using different hybrid - non-conventional energy resources are very encouraging and effective for future power supply in an Indian scenario. The comparative studies for different

power delivery models utilizing different sets of hybrid DERs depict different costs and environmental impacts. It has been observed that though case III provides a cost effective power delivery system, but from environment perspective, it has been obtained as an inferior system in comparison with case I. The power optimization using all the three types of DERs focuses on both economic and environmental aspects. Hence, the simulation models give us various power delivery options leading to more economical, eco-friendly and reliable future power. Moreover future studies can be carried out considering several constraining functions such as power interruption constraints and distribution losses. With more variations in load demand application of more enhanced optimization technique [17,18] can be made with better optimized results relating to both environmental as well as economic issues in distributed power system. Hybrid generation technologies involving sources such as micro-hydro and small-scale wind power, which are considered more autonomous and more sustainable than diesel generators [40], and which are especially suitable for rural electrification in developing countries can be investigated using soft computing techniques.

References

1. Jhonson, R.A., Agrawal, A.N., Chubb, T.J.: Simulink Model for Economic Analysis and En-vironmental Impacts of a PV With Diesel-Battery System for Remote Villages. *IEEE Transactions on Power System* 20(2), 692–700 (2005)
2. Fyfe, W.S., Powell, M.A., Hart, B.R., Ratanasthien, B.: A global crisis: Energy in the future. *Nonre-newable Resources*, 187–195 (1993)
3. Lasserter, R., Akhil, A., Marnay, C., Stephens, H., Dagle, J., Guttromson, R., Meliopoulos, A.S., Yinger, R., Eto, J.: The CERTS Microgrid Concept, White paper for CERTS (2002)
4. Bhattacharyya, S.C.: Energy access problem of the poor in India: Is rural electrification a remedy? *Energy Policy* 34(18), 3387–3397 (2006)
5. Hernandez-Aramburo, C.A., Green, T.C., Mugniot, N.: Fuelconsumption minimization of a microgrid. *IEEE Trans. Ind. Appl.* 41(3), 673–681 (2005)
6. Marnay, C., Venkataramanan, G., Stadler, M., Siddiqui, A., Firestone, R., Chandran, B.: Optimal Technology Selection and Operation of Commercial-Building Microgrids. *IEEE Trans. Power Systems* 23(3), 975 (2008)
7. Michiel, H., Petra, H., Bouwmans, I.: Socio-Technical Complexity in Energy Infrastructures Conceptual Framework to Study the Impact of Domestic Level Energy Generation, Storage and Exchange. In: *IEEE Int. Conf. Systems Man, and Cybernetics*, October 8-11 (2006)
8. Maribua, M., Firestone, K., Ryan, M., Marnay, C., Siddiquica, A.: Distributed energy resources market diffusion model. *Elsevier Energy Policy*, 4471–4484 (April 2007)
9. Patterson, W.D., Whitham, J.W.: *The Virtual Power Plant*, Standard & Poor's Utilities & Perspectives Special Technology Issue. McGraw-Hill Co. (1998)
10. Marin, F., Rey, A.B., Guerrero, A., de Ruz, F.A.: A future microgrid implementation based on renewable distributed resources for a clean green energy production. In: *Proc. IEEE PES Winter Meeting, CERTS*, pp. 305–308 (2002)
11. Miranda, V., Srinivasan, D., Proenca, L.M.: Evolutionary Computation in power system. *International Journal of Electric power & Energy System* 20(2), 89–98 (1998)

12. Farhat, I.A., EI-Hawary, M.E.: Optimization methods applied for solving the short-term hydrothermal co-ordination problem. *Electric Power System Research* 79(9) (September 2009)
13. Mandal, K.K., Basu, M., Chakraborty, N.: Particle swarm optimization – based fuzzy satisfying method for economic environmental dispatch of hydrothermal power systems. *International Journal of Automation and Control* (3), 216–239 (2009)
14. Chen-Ching, L., Dillon, T.: State of the art of expert system application to power systems. *International Journal of Electric Power & Energy System* 14(2-3), 95–96 (1992)
15. AlRashidi, M.R., EI-Hawary, M.E.: Application of computational intelligence techniques for solving the revived optimal power flow problem. *Electric Power System Research* 79(4), 694–702 (2009)
16. Marnay, C., Robio, F., Siddiqui, A.: Fuel Consumption Minimization of Microgrid. In: *Proc. IEEE PES Winter Meeting*, pp. 150–153 (2001)
17. Mallipeddi, R., Suganthan, P.N.: Efficient Constraint Handling for Optimal Reactive Power Dispatch Problem. *Swarm and Evolutionary Computation* 5, 28–36 (2012)
18. 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)
19. Meliopoulos, A.P.S.: Challenges in simulation and design of grids. In: *Proc. IEEE PES Winter Meeting*, pp. 309–314 (2002)
20. Marin, F., Rey, A.B., Guerrero, A., de Ruz, F.A.: A future microgrid implementation based on renewable distributed resources for a clean green energy production. In: *Proc. IEEE PES Winter Meeting, CERTS*, pp. 305–308 (2002)
21. Marnay, C., Robio, F., Siddiqui, A.: Fuel Consumption Minimization of Microgrid. In: *Proc. IEEE PES Winter Meeting*, pp. 150–153 (2001)
22. Meliopoulos, A.P.S.: Challenges in simulation and design of grids. In: *Proc. IEEE PES Winter Meeting*, pp. 309–314 (2002)
23. Chen, C., Duan, S., Cai, T., Liu, B., Hu, G.: Smart Energy Management System for Optimal Microgrid Economic Operation. *IET Renewable Power Generation* 5(3), 258–267 (2011)
24. Cooper, K., Dasgupta, A., Kennedy, K., Koelbel, C.: New Grid Scheduling and Re-Scheduling Methods in GRADs Project. In: *Proceedings on 18th International Symposium on Parallel and Distributed Processing*
25. Shi, R., Cui, C., Su, K., Zain, Z.: Comparison Study of Two Meta-Heuristic Algorithms with their Applications to Distributed Generation Planning. *Energy Procedia* 12, 245–252 (2011)
26. Chen, C., Duan, S., Cai, T., Liu, B., Hu, G.: Optimal Allocation and Economic Analysis of Energy storage System in Microgrids. *IEEE Transactions on Power Electronics* 26(10), 2762–2773
27. Niknam, T., Golestaneh, F., Shafiei, M.: Probabilistic Energy Management of a Renewable Microgrid with Hydrogen storage using Self adaptive Charge Search Algorithm. *Energy* (2012)
28. Zoka, Y., Sugimoto, A., Yorino, N., Kawahara, K., Kubokawa, J.: An economic evaluation for an autonomous independent network of distributed energy resources. *Electric Power System Research* 77(7), 831–838 (2007)
29. *Stand-Alone Photovoltaic Systems: A Handbook of Recommended Design Practices (Revised)*, Sandia National Labs, Albuquerque (1995)
30. Dawson, F.P., Dewan, S.B.: Remote diesel generator with photovoltaic cogeneration. In: *Proc. Solar Conf., Denver, CO*, pp. 269–274 (September 1989)

31. Farhat, F.A., Simoes, M.G.: *Itegration of Alternative Sources of Energy*. John Wiley & Sons, Inc. (2006)
32. Wikipedia contributors. List of countries by greenhouse gas emissions, http://en.wikipedia.org/wiki/List_of_countries_by_greenhouse_gas_emissions
33. http://www.retscreen.net/ang/emission_factors_for_diesel_generator_image.php
34. Lampinen, J.: A Constraint Handling Approach for Differential Evolution Algorithm. In: Proc. of the Congress on Evolutionary Computation, vol. 2, pp. 1468–1473 (2002)
35. Mezura–Montes, E., Velazquez–Reyes, J., Coello Coello, C.A.: Modified Differential Evolution of Constrained Optimization. IEEE Congress on Evolutionary Computation Sheraton (July 2006)
36. Bhattacharya, A.B., Kar, S.K., Bhattacharya, R.: Diffuse Solar Radiation and associated meteorological pa-rameters in India. *Annales Geophysicae* 14(10), 1051–1059 (1994)
37. Kolhe, M., Kolhe, S., Joshi, J.C.: Economic viability of stand-alone solar photovoltaic system in comparison with diesel-powered system for India. *Energy Economics* 24(2), 155–165 (2002)
38. Hunt, G.W.: Operational experience and performance characteristics of a valve-regulated lead–acid bat-tery energy-storage system for providing the customer with critical load protection and energy-management benefits at a lead-recycling plant. *Journal of Power Sources* 78, 171–175 (1999)
39. Dvoskin, D., Heady, E.O.: Commodity Prices and Resource Use Under Various Energy Alter-natives in Agriculture. *Western Journal of Agricultural Economics*, 53–62 (1977)
40. Private communication: E & M State Head Office, IIT kharagpur, West Bengal, India
41. Hybrid mini-grids for rural electrification: lessons learned. USAID: Alliance for rural electrification, http://www.ruralelec.org/fileadmin/DATA/Documents/06_Publications/Position_papers/ARE_Mini-grids_-_Full_version.pdf

Parameter Adaptation in Differential Evolution Based on Diversity Control

S. Miruna Joe Amali^{1,*}, and Subramanian Baskar²

¹ KLN College of Engineering

² Thiagarajar College of Engineering, Madurai

fausta_miruna@rediffmail.com, sbeee@tce.edu

Abstract. This paper explains an improved Differential Evolution algorithm based on adaptation of crossover rate and scaling factor using diversity control. Local search is applied to aid convergence process. The mutation strategies involved are modified using random localized method of vector selection to enhance performance. The proposed methodology is applied to SaDE. The proposed Diversity Controlled Parameter adapted Differential Evolution with Local Search (DCPaDE-LS) harmonically coordinates a balance between global and local search, thus ensuring a diversity dynamic which guarantees fast and efficient improvements in the search until detection of a solution with high performance. The performance of the proposed DCPaDE-LS is compared on a set of 26 bound-constrained benchmark functions for 10 and 30 dimensions with respect to average function evaluations (NFE) and success rate (SR) in 30 independent trials. Results show that, proposed method gives better SR for high-dimensional multimodal functions and saving in NFE for most functions.

Keywords: Differential evolution, diversity control, parameter adaptation, fuzzy system, local search.

1 Introduction

In Differential Evolution (DE) [1] there exist many trial vector generation strategies and three crucial control parameters i.e., population size, scaling factor (F), and crossover rate (CR), which significantly influence the optimization performance. Adaptation of the control parameters of DE has been an active research area. A complete survey of the various adaptation methods and future research trends in DE are given by [2]. Liu and Lampinen [3] use fuzzy logic controller to adapt F and CR in their fuzzy adaptive differential evolution (FADE). In [4] a uniform distribution between 0.5 and 1.5 for F adaptation with a mean value of 1 is employed. In [5] a self-adaptive parameter control similar to ES is implemented in jDE. SaDE [6] uses its previous learning experience to adaptively select mutation strategy as well as the associated control parameter values, which are often problem dependent. CR in SaDE is normally distributed in a range with mean CRm_k with respect to the k^{th} strategy

* Corresponding author.

and standard deviation 0.1. CRm_k is then updated by previous learning experience after an initial LP generations. F in SaDE is approximated by a normal distribution with mean value 0.5 and standard deviation 0.3.

In addition, the evolutionary process also depends on the population diversity. Specifically, higher dimensional, multimodal problems tend to suffer from loss of diversity during the earlier generations itself, leading to premature convergence. Controlling diversity help to maintain the explore-exploit cycle as proved in [7]. A parameter adaptation for DE (ADE) based on the idea of controlling the diversity using multi-population approach is reported in [8]. Mallipeddi and Suganthan [9] proposed the EPSDE algorithm, in which a pool of distinct mutation and crossover strategies coexists along with a pool of values for each control parameter throughout the evolution process and competes to produce offspring. A new mutation strategy named the greedy DE/current-to-best/1 mutation scheme and a fitness induced parent selection scheme is proposed for the binomial crossover of DE [10].

The diversity control methods available in literature mainly tend to avoid diversity loss or aid to improve the diversity level of an algorithm. Whereas, maintaining a balance between the conflicting goals of exploration and exploitation becomes a necessity for the identification of the global optimum. A diversity based crossover rate adaptation method using fuzzy systems is given in [11]. The methodology also includes a local search for a balanced explore-exploit cycle. This paper presents an improved DE which maintains a balance between explore-exploit cycles through diversity control and local search. This methodology is applied to SaDE [6]. The mutation strategies of SaDE have also been modified to improve the convergence speed by using a random localized method of vector selection [12].

2 Proposed Methodology

The DE algorithm optimizes a problem by generating a population of candidate solutions randomly and then applying mutation, crossover and selection operators to generate a new set of candidate solutions. In the original DE algorithm, there are only three control parameters, but the influence they have on the optimization problem is large. In addition, the selection of the mutation strategy also becomes a control parameter because of its effect on optimization. These control parameters maintain a desired diversity for the evolution to progress. In the proposed method a diversity control technique based on CR and F adaptation is performed through a feedback loop using fuzzy systems (FS). In the proposed work, two FS's are used one each for the adaptation of CR and F . A local search is also included to aid exploitation process. The block diagram of control parameter adaptation is given in Fig. 1. The two fuzzy systems modify the mean of the corresponding normal distribution (CR and F) to bring about the required changes in the control parameters. The diversity of the population is given as feedback for setting the values of the control parameters for the next generation. At the end of each generation, the diversity of the current population is calculated and the difference from the reference diversity (ref) is calculated. This difference is given as the feedback control to the FS. The FS outputs the modified

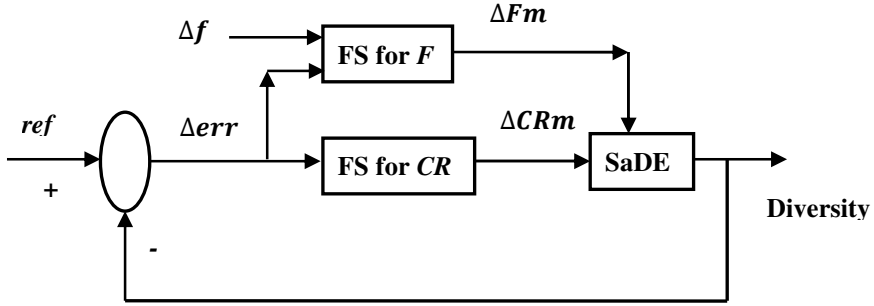


Fig. 1. Control parameter adaptation of SaDE using FS

CRm and Fm . This is then used to generate the crossover rate and scaling factor for the next generation of evolutionary operators. Using this methodology the Diversity Controlled Parameter adapted Differential Evolution with Local search (DCPaDE-LS) algorithm is proposed.

2.1 Diversity Measure

In the proposed methodology, diversity of the population is calculated using the ‘distance-to-average-point’ [7] measure defined as,

$$div_g(P) = \frac{1}{|L|*NP} * \sum_{i=1}^{NP} \sqrt{\sum_{j=1}^D (x_{ij} - \bar{x}_j)^2} \tag{1}$$

where $|L|$ is the length of the diagonal in the search space $S \subseteq R^D$, given by $\sqrt{\sum (x_{max} - x_{min})^2}$ with each search variable x in a finite range $x_{min} < x < x_{max}$, P is the population of size NP and D is the dimension of the problem, x_{ij} is the j^{th} value of the i^{th} individual and \bar{x}_j is the j^{th} value of the average point \bar{x} . The ‘distance-to-average-point’ measure considers the population size, dimensionality of the problem and the search range of each variable. Hence, this method is more suitable for a wide range of problems with varying characteristics.

2.2 Fuzzy System (FS)

Fuzzy system is the process of formulating a mapping, from a given input to an output using fuzzy logic. It is a method that interprets the values in the input vector and based on some set of rules, assigns values to the output vector. The mapping then provides a basis from which decisions can be made. The input and output variables are defined as fuzzy sets. A fuzzy set is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership. The degree an object belongs to a fuzzy set is denoted by a membership value between 0 and 1. A membership function associated with a given fuzzy set maps an input value to its appropriate membership value. It is a curve that defines how each point in the input space is mapped to a membership value. Fuzzy logic has been successfully applied in

diverse fields such as automatic control, data classification, decision analysis, expert systems, and computer vision. Mamdani's fuzzy system [13] is the most commonly used fuzzy methodology. Fuzzy logic process comprises of five parts: fuzzification of the input variables, application of the fuzzy operator (AND or OR) in the antecedent, implication from the antecedent to the consequent, aggregation of the consequents across the rules, and defuzzification.

Fuzzy System for Crossover Rate.

The CRm adaptation in the proposed methodology involves two components as given in Equation (2).

$$CRm_k = CRm_{s,k} + CRm_{d,k} \tag{2}$$

where $CRm_{s,k}$ is obtained from SaDE which considers the median of all the successful CR values in the past LP generations and $CRm_{d,k}$ component is based on diversity. In order to establish a stable diversity control, the FS is employed as it has been successfully applied in diverse decision making and control applications. The FS will adapt the CRm value automatically without any user intervention depending upon the characteristics of the problem. In the proposed work, a single input, single output Mamdani's fuzzy logic system is used. The input to the FS is the diversity error Δerr_g defined as the difference between a reference diversity (ref) and diversity div_g given in Equation (3).

$$\Delta err_g = ref - div_g \tag{3}$$

The output of the FS is the change in mean of the normal distribution for crossover rate (ΔCRm_k^g) for k^{th} strategy. The CRm_k^g for the g^{th} generation is as given in Equation (4).

$$CRm_k^g = CRm_k^{g-1} + \Delta CRm_k^g \tag{4}$$

Both the input and output of the FS are defined as fuzzy variables. They are divided into five sub functions namely, Large Negative (LN), Negative (N), Zero (Z), Positive (P) and Large Positive (LP) each defined by triangular membership functions. The centroid method is used for defuzzification. The rule set for the FS is given in Table 1. The rule set is framed on the basis that when Δerr_g is positive (P), it implies that the reference diversity (ref) is greater than diversity div_g , so in order to increase the diversity, the CR_g value has to be increased. Hence, for every positive value of Δerr the output of FS (ΔCRm_g) should also be positive. Similarly, when the Δerr_g value is Large Negative (LN), it implies that the diversity div_g is very large. Thus, to decrease the diversity, the CR_g value should also be decreased, that is the ΔCRm_g value should be Large Negative (LN). As there is overlapping among the membership functions defined for Δerr , at any point of time more than one rule being fired and correspondingly defuzzification gives an overall better ΔCRm value. Thus, there is a change in CR_g value which maintains a better diversity. This is achieved for all the three mutation strategies with crossover.

Table 1. Rule set for crossover rate Fuzzy System

| | | | | | |
|--------------|----|---|---|---|----|
| Δerr | LN | N | Z | P | LP |
| ΔCRm | LN | N | Z | P | LP |

Fuzzy System for Scaling Factor.

The next important parameter namely, the scaling factor F controls the amplification of the difference vectors. As the value of the difference vector decreases, the perturbation also decreases i.e. as the population gets closer to the optimum the step length is automatically decreased. During the early stages of evolution when a more diversified search is required a larger value of F is optimal, but during the later stages as the search gets more intensified a smaller value of F is beneficial. The mutation operator is applied to each target vector and it depends on the location of the target vector in the search space. The mutation operator depends on the positioning of the individuals and directly affects diversity. Hence, both these parameters are given as input to the FS. A two input, single output Mamdani’s FS is used for the adaptation of F . The inputs to the FS are Δerr_g given in Equation (3) and the absolute difference in the objective values between the best individual and the i^{th} individual given in Equation (5).

$$\Delta f_g^i = | f_g(x_{best}) - f_g(x_i) | \tag{5}$$

where $f(.)$ is the objective function. The output of the FS is the change in the mean of the normal distribution ΔFm_g^i of F for all the individuals in the population. The mean of the normal distribution of F for the i^{th} individual is given in Equation (6).

$$Fm_g^i = Fm_{g-1}^i + \Delta Fm_g^i \tag{6}$$

In this way, for the individuals away from the best, larger values of F is generated, which aid searching different regions of the search space. At the same time, for individuals closer to the current best, smaller values of F is generated, which aid in fine tuning the search in order to identify the global optimum. The Δerr_g input is formed similar to the FS for CR adaptation. The Δf_g^i input is divided into Z, P and LP within $[0 1]$, as it is an absolute value. The ΔFm output of the FS is defined by LN, N, Z, P and LP. All the linguistic variables are represented by triangular membership functions. The centroid method is used for defuzzification.

The rule set for the FS is given in Table 2. The rule set is framed on the basis that when the Δerr_g input is LP and Δf_g^i is also LP, it implies that the reference diversity (ref) is greater than diversity div_g and the individual is far away from the best individual of the current population. Hence, in order to explore wider regions of the search space and to enhance the diversity the output of the FS i.e. the mean of the normal distribution for scaling factor Fm_g^i should also be increased with an output value of LP. Similarly, when the Δerr_g input is N and Δf_g^i is also Z, it implies that the reference diversity is lesser than the diversity of the population and the individual is in the close proximity of the best individual in the current population. Hence, in

order to search finer regions of the search space and improve exploitation the diversity must be decreased. Thus the mean of the normal distribution for scaling factor Fm_g^i should be marginally decreased with an output value of N. The complete rule set with 15 rules is framed on this ideology.

Table 2. Rule set for scaling factor FS

| | | | | | |
|----------------------------------|----|---|----|---|----|
| $\Delta f \backslash \Delta err$ | LN | N | Z | P | LP |
| LP | LN | N | Z | P | LP |
| P | LP | N | Z | N | LN |
| Z | LN | N | LN | N | Z |

2.3 Local Search

Parameter adaptation based on the diversity tends to maintain a desired diversity level. This diversity control will distract the population from convergence as the evolution progresses. During later evolutionary stages, diversity must be compromised to aid in convergence for the exploitation of useful information in the population. Thus, for exploitation to occur the diversity must be decreased. Local search is a well known method for exploitation [14]. Local search tries to identify a better solution in the proximity of a chosen individual. The search takes place locally within the bounded proximity of the chosen individual rather than in a large diverse region. The locally improved individual will be placed in the population to compete with others in the next generation. This reduces the number of evaluations required to identify the global optimum. Otherwise, the population will be diverse for more number of generations and convergence could take a long time or the population could be too diverse so as not to converge at all. In order to maintain a balance between explore and exploit cycle, a local search method becomes a necessity.

In the proposed methodology, local search is implemented once every $Lcnt$ generation with a predefined $Lfeval$ number of function evaluations at the best individual. Local search at the best individual has a higher probability of escaping from a local optimum, hence this method is adapted. When an individual better than the current best solution is identified, a random individual is replaced with the locally better solution. This local search with elitism will guide the population to the global optimum, thereby improving exploitation. Local search is done only for every $Lcnt$ generation to avoid computational overhead. The local search evaluations must be controlled to avoid over exploitation, which could also lead to premature convergence. Thus a controlled local search methodology with predefined number of local evaluations is implemented.

3 Experiment Setup and Result Analysis

In order to demonstrate the effects of parameter adaptation based on diversity control and local search the following algorithm are considered for comparison.

1. SaDE [6]
2. SaDE with Local Search (SaDE-LS)
3. Diversity Controlled SaDE with Local Search (DCSaDE-LS) [11]
4. Diversity Controlled Parameter adapted DE with Local Search (DCPaDE-LS)

In order to demonstrate the effects of the proposed methodology, the DCPaDE-LS algorithm is compared with SaDE, SaDE with local search (SaDE-LS) and Diversity controlled SaDE with local search (DCSaDE-LS) [11]. The algorithms are implemented using MATLAB 7.9 using Intel Core 2 CPU, operating at 1.86 GHz with 2GB RAM. The test suite of 26 unconstrained functions as reported in [6] is used for testing the performance. In order to have a fair comparison of results, same number of function evaluations and population size as in [6] are employed. The CR and F are generated in the range (0, 1) and (0.4, 0.8) respectively. The ref , $Lcnt$ and $Lfeval$ values are set using trial and error method to avoid additional overhead. An extensive study was performed in order to determine the effects of the various control parameters. The $Lcnt$ and $Lfeval$ does not have distinct impact on the convergence characteristics of the problems, whereas, they affect the speed of convergence (NFE). Thus, a nominal value is set for the parameters. ref and $Lcnt$ are set at 0.15 and 30 generations respectively. $Lfeval$ is set at 300 for 10D and 30D. $Lfeval$ is set at $(10 * D)$ for $f_{15} - f_{26}$ as the function dimension varies.

To verify the consistency of the algorithms, they are compared with respect to success rate (SR) and average number of function evaluations (NFE) in 30 trials. Where, SR is defined as the ratio between successful trials to the total number of trials. A successful trial is one which results in a function value no worse than the predefined optimal value, checked to the order of $1e-5$ for all the test functions ($f(x^*) + 1e - 5$, where x^* is the global optimum) with a number of function evaluations within the predefined maximum value. The function evaluations used during the local search are also included in the overall NFE.

3.1 10D Results

The SR/NFE and % change in NFE for 10D is given in Table 3. In those functions with 100% SR the NFE is reported. A negative % change indicates increase in NFE as compared to SaDE. The table shows that there is no much difference in SR between the algorithms in all the functions except f_3 , f_8 and f_{14} . SaDE-LS algorithm includes only the local search methodology without incorporating the diversity controlled parameter adaptation. The SaDE-LS is able to produce 96.6% and 13.3% SR for f_3 and f_8 respectively which is marginally lower than SaDE performance. However, DCPaDE-LS is able to produce only 0% SR for f_8 . In all other functions, DCPaDE-LS does not report any deterioration in SR. All the algorithms report an improved SR for f_{14} compared to SaDE. The DCPaDE-LS achieves lower NFE than SaDE for all the functions except f_7 . It reports least NFE in 6 functions, whereas SaDE-LS algorithm reports the least NFE in 4 functions. The NFE of DCPaDE-LS is better than DCSaDE-LS in 9 functions. The DCPaDE-LS gives a minimum of 20% and a maximum of 88%

decrease in NFE in those functions with better performance. This clearly shows that the introduction of adaptation of CR and F parameters along with diversity control is able to distinctly reduce the NFEs. But SaDE-LS outperform DCPaDE-LS in terms of NFE on five functions f_5, f_6, f_7, f_9 and f_{11} . This proves that having a local search aids in the convergence of the population, but it can also lead to premature convergence if not properly governed by a diversity control measure.

Table 3. Comparison of SR/NFE for 10D functions

| 10 # | SaDE | SaDE-LS | | DCSaDE-LS | | DCPaDE-LS | |
|----------|--------------|--------------|----------|--------------|----------|--------------|----------|
| | NFE | NFE | % change | NFE | % change | NFE | % change |
| f_1 | 8375 | 1586 | 81.06 | 1593 | 80.78 | 1579 | 81.15 |
| f_2 | 14867 | 2001 | 86.54 | 1702 | 88.55 | 1691 | 88.63 |
| f_3 | 42446 | 96.6% | NA | 11862 | 72.05 | 11361 | 73.23 |
| f_4 | 15754 | 18280 | -16.03 | 28346 | -79.93 | 12295 | 21.96 |
| f_5 | 12123 | 7198 | 40.62 | 9678 | 20.17 | 8196 | 32.39 |
| f_6 | 12244 | 7134 | 41.73 | 11371 | 7.13 | 8286 | 32.33 |
| f_7 | 35393 | 40499 | -14.43 | 48328 | -36.55 | 65952 | -86.34 |
| f_8 | 20% | 13.3% | NA | 20% | NA | 0% | NA |
| f_9 | 23799 | 17383 | 26.96 | 18712 | 21.37 | 18842 | 20.83 |
| f_{10} | 0% | 0% | NA | 0% | NA | 0% | NA |
| f_{11} | 26945 | 19637 | 27.12 | 23002 | 14.63 | 20279 | 24.74 |
| f_{12} | 16663 | 9287 | 44.27 | 8974 | 46.14 | 6852 | 58.88 |
| f_{13} | 9740 | 2229 | 77.11 | 5721 | 41.26 | 1897 | 80.52 |
| f_{14} | 80% | 86.6% | NA | 93.3% | NA | 93.3% | NA |

NA - Not Applicable (Problems with less than 100% SR)

3.2 30D Results

The SR and NFE for 30D $f_1 - f_{14}$, for the algorithms in comparison is given in Table 4. DCPaDE-LS shows improvement in terms of SR in f_3, f_4, f_7 and f_8 functions compared to SaDE. The SR of SaDE-LS has reduced for $f_5, f_6, f_7, f_8, f_9, f_{11}$ and f_{12} . This is due to rapid loss of diversity which results in premature convergence. This proves that though local search is able to reduce the NFE is few 10D functions, for higher dimensional functions the SaDE-LS fails due to loss of diversity. The SR of DCPaDE-LS is better than DCSaDE-LS in f_4, f_7 and f_8 functions. The diversity characteristics over generations of the algorithms for 30D f_7 is given in Fig. 2. It can be seen that for about 300 generations the diversity of DCPaDE-LS is better than other algorithms. This improved diversity aids the algorithm to escape local optimum. The diversity controlled parameter adaptation provides the necessary variation in CR_m and F_m values. Due to the improved diversity, the convergence of the DCPaDE-LS takes more number of generations for few complex functions. This can be seen in the convergence characteristics for 30D f_7 in Fig. 3(a). It is given in semilog scale at the Y- axis for more clarity. It can be observed that the SaDE-LS and SaDE converge around 820 and 880 generations, whereas the DCPaDE-LS convergences around 1090 generation.

Though it requires more number of generations, the DCPaDE-LS identifies the optimal solution in all the 30 trials. In few other functions DCPaDE-LS shows savings in NFE.

Table 4. Comparison of SR/NFE for 30D functions

| 30 | SaDE | SaDE-LS | | DCSaDE-LS | | DCPaDE-LS | |
|----------|--------------|--------------|----------|-----------|----------|--------------|----------|
| # | NFE | NFE | % change | NFE | % change | NFE | % change |
| f_1 | 20184 | 1784 | 91.16 | 1803 | 91.07 | 1768 | 91.24 |
| f_2 | 118743 | 16175 | 86.38 | 99155 | 16.50 | 16621 | 86.0 |
| f_3 | 90% | 90% | NA | 101882 | NA | 76395 | NA |
| f_4 | 0% | 0% | NA | 3.3% | NA | 30% | NA |
| f_5 | 26953 | 93.3% | NA | 21887 | 18.79 | 17757 | 34.12 |
| f_6 | 33014 | 86.6% | NA | 24650 | 25.33 | 17617 | 46.64 |
| f_7 | 80% | 90% | NA | 93.3% | NA | 30479 | NA |
| f_8 | 40% | 26.6% | NA | 53.3% | NA | 43937 | NA |
| f_9 | 58723 | 60% | NA | 75007 | -27.73 | 76078 | -29.55 |
| f_{10} | 0% | 0% | NA | 0% | NA | 0% | NA |
| f_{11} | 77920 | 66.6% | NA | 94886 | -21.77 | 91454 | -17.37 |
| f_{12} | 44283 | 93.3% | NA | 34022 | 23.17 | 31536 | 28.78 |
| f_{13} | 19031 | 6574 | 66.46 | 14501 | 23.80 | 5567 | 70.75 |
| f_{14} | 0% | 0% | NA | 0% | NA | 0% | NA |

NA – Not Applicable (Functions with less than 100% SR)

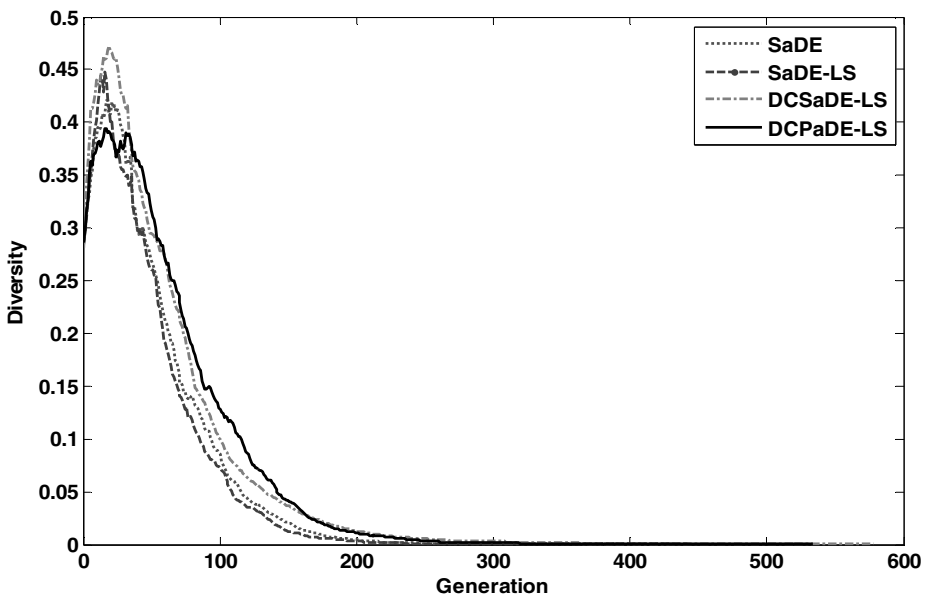


Fig. 2. Diversity characteristics of all the algorithms for 30D f_7

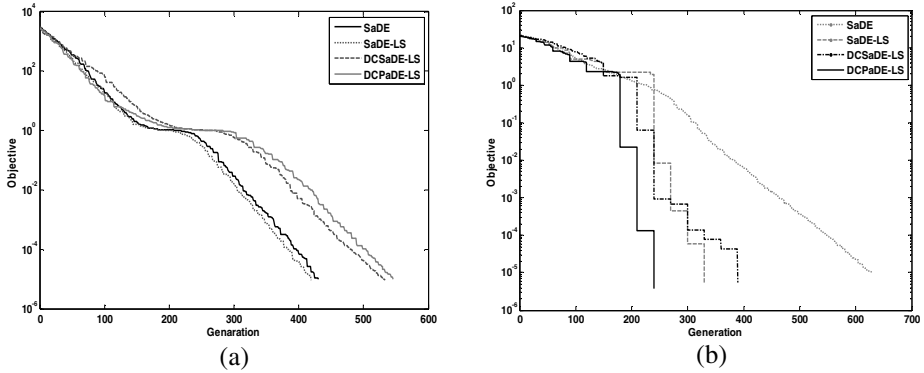


Fig. 3. Convergence characteristic of all the algorithms for 30D (a) f_7 and (b) f_6

The convergence characteristics of the algorithms for f_6 are given in Fig. 3(b). It is clear from the figure that DCPaDE-LS converges much faster around 250 generation. Except for f_9 and f_{11} DCPaDE-LS requires only less NFE compared to that of SaDE. The SaDE-LS reports the least NFE in f_2 . Compared to DCSaDE-LS, the DCPaDE-LS reports lower NFE in all the functions except f_9 . The percentage decrease in NFE achieved by DCPaDE-LS is in the range of 28% to 91% for those functions with better performance. Results clearly show that with the addition of parameter adaptation method based on diversity control and a local search method improves convergence and efficiency for multimodal problems.

The algorithms are also tested on functions with variable dimensions $f_{15} - f_{26}$. The SR/NFE and % change in NFE with respect to SaDE is given in Table 5.

Table 5. Comparison of NFE for functions $f_{15} - f_{26}$

| # | SaDE | SaDE-LS | | DCSaDE-LS | | DCPaDE-LS | |
|----------|--------------|--------------|----------|-----------|----------|--------------|----------|
| | NFE | NFE | % change | NFE | % change | NFE | % change |
| f_{15} | 25137 | 22617 | 10.02 | 24320 | 3.25 | 21061 | 16.21 |
| f_{16} | 88934 | 119572 | -34.45 | 206280 | -131.94 | 74860 | 15.82 |
| f_{17} | 18742 | 19464 | -3.85 | 20498 | -9.37 | 30017 | -60.16 |
| f_{18} | 19390 | 15144 | 21.90 | 15639 | 19.35 | 20417 | -5.30 |
| f_{19} | 6426 | 6226 | 3.11 | 7346 | -14.32 | 3859 | 39.95 |
| f_{20} | 2076 | 1522 | 26.69 | 1523 | 26.64 | 1494 | 28.03 |
| f_{21} | 2614 | 1565 | 40.13 | 1567 | 40.05 | 1515 | 42.04 |
| f_{22} | 802 | 850 | -5.98 | 786 | 1.99 | 687 | 14.34 |
| f_{23} | 3080 | 2431 | 21.07 | 3028 | 1.69 | 2280 | 25.97 |
| f_{24} | 4947 | 3434 | 30.58 | 3926 | 20.64 | 2857 | 42.24 |
| f_{25} | 4173 | 3121 | 25.21 | 3834 | 8.12 | 2601 | 37.67 |
| f_{26} | 4267 | 3008 | 29.51 | 2657 | 37.73 | 2439 | 42.84 |

Except for f_{17} and f_{18} , DCPaDE-LS is able to provide 100% SR with less NFE. The SaDE-LS reports the least NFE in f_{18} . The NFE reported by DCPaDE-LS is better than DCSaDE-LS in all the functions except f_{17} and f_{18} . The NFE decrease is in the

range of 14% to 42% for DCPaDE-LS. The proposed DCPaDE-LS algorithm harmonically coordinates a balance between global and local search. This ensures a diversity dynamic which guarantees fast and efficient improvements in the search until detection of a solution with high performance.

4 Conclusions

The proposed algorithm presents an improved diversity control based parameter adaptation method using fuzzy logic. This methodology has been applied to SaDE algorithm to adapt CR and F . Diversity control is realized with the help of two separate FS, one each for CR and F . It also includes a local search at regular intervals to aid in convergence and named as Diversity Controlled Parameter adaptation in DE with Local Search (DCPaDE-LS). The simulation results show that there is no much difference in performance in terms of SR for 10D functions among the algorithms, but the DCPaDE-LS algorithm gives savings in terms of NFE for most of the functions. DCPaDE-LS algorithm shows improvement in terms of SR as well as NFE for 30D functions. In f_{15} - f_{26} , DCPaDE-LS gives improvement in terms of NFE is almost all the functions. This proves the efficiency of the algorithm on higher dimensional multimodal functions due to diversity control and local search in tandem. The efficiency of the proposed algorithm is attributed to the balanced explore and exploit cycles. Since the parameters are adapted using FS, rule based manual tuning of parameters is not required. This further enhances the applicability of the algorithm. The diversity based parameter adaptation using FS with local search methodology can be easily incorporated to any evolutionary algorithm to improve its performance.

References

1. Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Opt.* 11, 341–359 (1997)
2. Das, S., Suganthan, P.N.: Differential Evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* 15(1), 4–31 (2011)
3. Liu, L., Lampinen, J.: A fuzzy adaptive differential evolution algorithm. *Soft Comput.* 9(6), 448–462 (2005)
4. Chakraborty, U.K., Das, S., Konar, A.: Differential evolution with local neighborhood. In: *Proc. Congr. Evolut. Comput.*, Vancouver, BC, Canada, pp. 2042–2049 (2006)
5. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* 10(6), 646–657 (2006)
6. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization. *IEEE Trans. Evol. Comput.* 13(2), 398–417 (2009)
7. Ursem, R.K.: Diversity-Guided Evolutionary Algorithms. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) *PPSN 2002. LNCS*, vol. 2439, pp. 462–471. Springer, Heidelberg (2002)

8. Zaharie, D.: Control of population diversity and adaptation in differential evolution algorithms. In: Proc. Mendel 9th Int. Conf. Soft Comput., Brno, CR, pp. 41–46 (2003)
9. Islam, S.M., Das, S., Ghosh, S., Roy, S., Suganthan, P.N.: An Adaptive Differential Evolution Algorithm with Novel Mutation and Crossover Strategies for Global Numerical Optimization. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics* 42(2), 482–500 (2012)
10. Mallipeddi, R., Suganthan, P.N.: Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing* 11(2), 1679–1696 (2011)
11. Miruna Joe Amali, S., Baskar, S.: Fuzzy logic based diversity controlled self adaptive differential evolution. *Engineering Optim.* (2012), doi:10.1080/0305215X.2012.713356
12. Kaelo, P., Ali, M.M.: A numerical study of some modified differential evolution algorithms. *Eur. J. Oper. Res.* 169, 1176–1184 (2006)
13. Jang, J.S.R., Sun, C.T., Mizutani, E.: *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall (1996)
14. Huang, T., Huang, J., Zhang, J.: An orthogonal local search genetic algorithm for the design and optimization of power electronic circuits. In: *IEEE Cong. Evol. Comput.*, Hong Kong, pp. 2452–2459 (2008)

Data Clustering with Differential Evolution Incorporating Macromutations

Goran Martinović and Dražen Bajer

Faculty of Electrical Engineering,
J. J. Strossmayer University of Osijek,
Kneza Trpimira 2b, 31000 Osijek, Croatia
{goran.martinovic,drazen.bajer}@etfos.hr

Abstract. Data clustering is one of the fundamental tools in data mining and requires the grouping of a dataset into a specified number of nonempty and disjoint subsets. Beside the usual partitional and hierarchical methods, evolutionary algorithms are employed for clustering as well. They are able to find good quality partitions of the dataset and successfully solve some of the shortcomings that the k-means, being one of the most popular partitional algorithms, exhibits. This paper proposes a differential evolution algorithm that includes macromutations as an additional exploration mechanism. The application probability and the intensity of the macromutations are dynamically adjusted during runtime. The proposed algorithm was compared to four variants of differential evolution and one particle swarm optimization algorithm. The experimental analysis conducted on a number of real datasets showed that the proposed algorithm is stable and manages to find high quality solutions.

Keywords: Data clustering, Davies-Bouldin index, differential evolution, macromutations, representative points.

1 Introduction

Cluster analysis or clustering [1], [2] represents the unsupervised classification of data and it is one of the fundamental methods in data mining. Clustering requires the division of a given dataset into a specified number of nonempty and disjoint subsets such that data/patterns belonging to the same subset are similar according to a measure and that data belonging to different subsets is dissimilar according to the same measure. The application area of clustering includes the analysis of gene expression data, image segmentation, vector quantization etc.

According to [2], clustering methods can be grouped into two classes; hierarchical and partitional. This paper focuses on (hard) partitional clustering, while an overview of hierarchical methods can be found e.g. in [3]. Partitional methods iteratively try to improve an initial partition by switching data between clusters.

One of the most significant and widely used partitional clustering methods is the k-means [4] algorithm. Although, it has some drawbacks, like sensitivity

to initial conditions and it easily gets stuck in local optima. To solve the aforementioned shortcomings evolutionary algorithms (EAs) or swarm intelligence algorithms may be used as they are able to efficiently and effectively explore the search space. A comprehensive overview of different EAs for data clustering can be found in [5].

Laszlo and Mukherjee [6] presented a genetic algorithm (GA) for data clustering that utilizes a region-based crossover operator. They use the GA to find good initial centroids for the k-means algorithm. The chromosomes in their GA encode the centroids of clusters, and during crossover a pair of chromosomes exchange a number of centroids that occupy the same region of space. According to the conducted experimental analysis, the region-based crossover is superior to a random exchange of centroids. A differential evolution (DE) algorithm incorporating the k-means algorithm was proposed by Kwedlo [7]. He used the k-means algorithm for generating the initial population and local search as well. The vectors of the population encoded the centroids of the clusters. Also, Kwedlo applies a reordering procedure to the vectors in an attempt to remove the redundancy inherent to the centroid encoding. Hatamlou *et al.* [8] presented a method that combines the gravitational search algorithm and k-means. Generation of the initial population was enhanced with k-means; one individual was created with k-means, additional three from the dataset itself, and the rest randomly. According to Hatamlou *et al.*, this should decrease the number of necessary iterations and enable the exploration of the promising parts of the search space. Chuang *et al.* [9] presented a data clustering algorithm based on a particle swarm optimization (PSO) algorithm that uses the Gauss chaotic map. The random factors that influence the cognitive and social component of the velocity update were substituted with sequences generated by the Gauss chaotic map. According to Chuang *et al.*, this should lead to a balance between exploration and exploitation in the search. Zou *et al.* [10] proposed a cooperative artificial bee colony (CABC) algorithm for data clustering. In the CABC algorithm all the bees (employed as well as on-looker) participate in generating a so-called super best solution. Every solution component (a centroid) of the super best solution is considered to be replaced by the corresponding solution component of each bee. The replacement takes place only if it yields a better solution. This way good components of the solutions found by the bees are preserved. The aforementioned approaches all adopted the centroid encoding for candidate solutions. A slightly different encoding was utilized by Paterlini and Krink [11]. For the representation of candidate solutions they utilized representative points that, generally, are not centroids, and a partition of the dataset was obtained, as in the case of centroids, by assigning data to the cluster associated with the nearest representative point. Paterlini and Krink conducted a comparison between DE, PSO, GA and random search for data clustering. The experimental analysis showed the superiority of the DE compared to the other algorithms employed by them.

This paper proposes a DE algorithm for data clustering. The algorithm incorporates macromutations as an additional genetic operator. Its purpose is to enable a more extensive exploration of the search space, especially when further

exploration with the common mutation and crossover is difficult because of the high similarity of the vectors in the population. The application probability and intensity of the macromutations are dynamically adjusted during runtime.

The remainder of the paper is organized as follows. Section 2 describes the problem of data clustering. Section 3 briefly describes the DE algorithm and its application to data clustering. The proposed algorithm is described in Sect. 4. The conducted experimental analysis is shown in Sect. 5.

2 Data Clustering

Clustering requires the division of a given dataset $\mathcal{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_n\} \subset \mathbb{R}^p$ where $n \geq 2$ into $1 \leq k \leq n$ disjoint subsets (clusters) c_1, \dots, c_k such that (1) holds.

$$\bigcup_{i=1}^k c_i = \mathcal{A}, \quad c_i \cap c_j = \emptyset, \quad \forall i \neq j, \quad |c_i| \geq 1, \quad i = 1, \dots, k \quad (1)$$

The division of a dataset \mathcal{A} into k subsets, c_1, \dots, c_k , such that (1) holds can be denoted $\mathcal{C} = \{c_1, \dots, c_k\}$ and represents a partition of \mathcal{A} . The number of clusters k may be application dependent and known a priori, but some times this is not the case and k must be determined in some way. In this paper, the number of clusters is assumed to be known a priori.

Since the number of all possible partitions of \mathcal{A} is extremely large (Stirling's number of second kind) a criterion must be defined to evaluate the different partitions. In the literature a large number of cluster validity criteria (indices) exist for evaluating possible partitions of a set. A comprehensive overview and comparison of different criteria may be found e.g. in [12]. One of the best known (relative) criteria is the Davies-Bouldin index [13] DB as given by Eq. (2).

$$DB = \frac{1}{k} \sum_{i=1}^k D_i . \quad (2)$$

In (2) $D_i = \max_{j \neq i} D_{ij}$ where $D_{ij} = (s_i + s_j)/h_{ij}$, and s_i i.e. s_j and h_{ij} are calculated as per (3) and (4), respectively.

$$s_i = \frac{1}{|c_i|} \sum_{\mathbf{a}_j \in c_i} d(\mathbf{a}_j, \mathbf{z}_i) . \quad (3)$$

$$h_{ij} = d(\mathbf{z}_i, \mathbf{z}_j) . \quad (4)$$

Equation (3) represents the dispersion of cluster c_i , and Eq. (4) the distance between cluster centroids \mathbf{z}_i and \mathbf{z}_j , where $\mathbf{z}_l = 1/|c_l| \sum_{\mathbf{a}_j \in c_l} \mathbf{a}_j$ for $l = 1, \dots, k$. In both (3) and (4) $d(\cdot, \cdot)$ represents a metric, usually the Euclidean distance which is also used in this paper.

Smaller values of the Davies-Bouldin index indicate better partitions composed of compact and separated clusters.

3 Differential Evolution

Differential evolution [14] is a simple, yet effective optimization and search method for global optimization problems. It maintains a population of vectors that represent candidate solutions to the problem at hand. Like other common EAs it utilizes crossover, mutation and selection to drive the search process.

The population of DE is composed of real-valued vectors $\mathbf{v}_i = (v_i^1, \dots, v_i^D) \in \mathbb{R}^D$ for $i = 1, \dots, NP$. The initial population is usually generated randomly. A new population of so-called trial vectors is generated in each generation through mutation and crossover. Mutation creates for every vector \mathbf{v}_i (target vector) as per (5) a corresponding mutant/donor which is used for crossover as per (6).

$$\mathbf{u}_i = \mathbf{v}_{r_1} + F \cdot (\mathbf{v}_{r_2} - \mathbf{v}_{r_3}) . \quad (5)$$

$$\mathbf{t}_i^j = \begin{cases} \mathbf{u}_i^j, & \text{if } \mathcal{U}[0, 1] \leq CR \text{ or } j = r_j \\ \mathbf{v}_i^j, & \text{else} \end{cases} , \text{ for } j = 1, \dots, D . \quad (6)$$

In (5) \mathbf{u}_i is the mutant/donor, \mathbf{v}_{r_1} , \mathbf{v}_{r_2} , and \mathbf{v}_{r_3} are randomly chosen vectors from the population satisfying the condition $i \neq r_1 \neq r_2 \neq r_3$; $F \in [0, \infty)$ is the scale factor and it controls the mutation step size. In (6) \mathbf{t}_i is the trial vector obtained by crossing over vectors \mathbf{v}_i and \mathbf{u}_i , while $\mathcal{U}[0, 1)$ is an uniform random number in the range $[0, 1)$, $CR \in [0, 1)$ is the crossover rate and r_j is a randomly chosen number from the set $\{1, \dots, D\}$.

Once the trial vector population is created selection takes place and a given trial vector replaces a corresponding target vector if it is equal or better than it.

The described algorithm represents the classical/canonical DE and is usually denoted DE/rand/1/bin [14]. Beside the classical DE, a popular DE strategy/variant is the DE/best/1/bin [15] where always the best vector in the current population is selected as the so-called base vector (\mathbf{v}_{r_1} in Eq. (5)). A comprehensive overview of different DE strategies as well as their application areas can be found in [15].

3.1 Differential Evolution for Data Clustering

As has already been mentioned, DE has been successfully applied for data clustering [7], [11]. The first step in designing an EA is the decision on a representation of candidate solutions. For data clustering, a common choice is the centroid encoding [6,7,8,9,10] where each individual represents a required number of cluster centroids, and a partition of a dataset is obtained by assigning data to the cluster associated with the nearest centroid. A different representation was used in [11] which is also adopted in this paper. Namely, instead of centroids, representative points are encoded, and a partition is obtained in the same way as for the centroid encoding. Generally the representative points may be any point in \mathbb{R}^p , but are usually constrained with the dataset itself – $[\mathbf{x}_{min}, \mathbf{x}_{max}]$, where \mathbf{x}_{min} and \mathbf{x}_{max} are vectors composed of the minimal and maximal values for each feature in the dataset.

For the aforementioned representation (also true for the centroid encoding) the DE vectors are real-valued and of dimensionality $D = k \times p$, where k is the number of clusters, while p represents the dimensionality (number of features) of the data (Fig. 1). Thus, there is no need to modify the mutation and crossover operators. Also, the problem of data clustering reduces to a global optimization problem. For evaluating the solutions the Davies-Bouldin index is used in this paper as given by Eq. (2), while according to (1) infeasible solutions are those where one or more clusters are empty. Infeasible solutions can be handled by a penalty function. Based on the aforementioned, in this paper, solutions are evaluated as per Eq. (7).

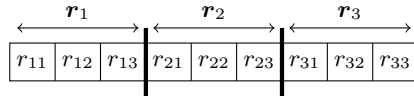


Fig. 1. Representative points encoding of candidate solution – 3 clusters and 3 features

$$f(\mathcal{C}) = \begin{cases} DB & , \text{ if (1) is satisfied} \\ g(\mathcal{C})^2 \cdot 10^5 & , \text{ else} \end{cases} . \quad (7)$$

According to (7), solutions i.e. the partitions they represent are evaluated with the Davies-Bouldin index if there are no empty clusters, else they are penalized; where $g(\mathcal{C})$ is the number of empty clusters. This way infeasible solutions are well separated from feasible ones.

4 Proposed Algorithm

When the population vectors become very similar or equal the exploration of the search space with mutation and crossover becomes very slow/difficult or impossible. This only leads to a waste of computational time since the discovery of better solutions is very unlikely. Thus, this paper proposes the incorporation of macromutations [16] (also referred to as the headless chicken operator) into the classical DE as an additional exploration mechanism. Macromutation represents the crossover of a given population vector \mathbf{v}_i with a randomly generated one (Eq. (8)). The crossover is controlled with the crossover rate p_c . The proposed algorithm, hereinafter referred to as DEMM, is based on DE/rand/1/bin and incorporates macromutations. A high-level outline of DEMM is given in Alg. 1.

$$\mathbf{m}_i^j = \begin{cases} \mathcal{U}[\mathbf{x}_{min}^{j \bmod p}, \mathbf{x}_{max}^{j \bmod p}] & , \text{ if } \mathcal{U}[0, 1) < p_c \text{ or } j = r_j \\ \mathbf{v}_i^j & , \text{ else} \end{cases} , j = 1, \dots, D . \quad (8)$$

In (8) \mathbf{m}_i is the D -dimensional vector obtained through macromutation. The crossover rate p_c is dynamically adjusted with the number of executed iterations

t as per Eq. (9). Also, in (8) p is the data dimensionality, while r_j is, as in (6), a randomly chosen number from the set $\{1, \dots, D\}$. The purpose of r_j is to ensure that at least one random parameter is introduced during the macromutation.

$$p_c(t + 1) = p_c(t) / {}^{t_{max}}\sqrt{p_c^{max} / p_c^{min}} . \tag{9}$$

The macromutations in DEMM are applied instead of the usual mutation and crossover. More precisely, the macromutations are applied with a set probability p_{MM} which is also dynamically adjusted with the number of executed iterations t as per Eq. (10).

$$p_{MM}(t + 1) = p_{MM}(t) + (p_{MM}^{max} - p_{MM}^{min}) / t_{max} . \tag{10}$$

In both (9) and (10) t_{max} represents the set (maximum) number of iterations to be executed. According to (9), the crossover rate (intensity) of the macromutations is exponentially decreased with the values of p_c^{max} and p_c^{min} being 0.5, and 0.005, respectively. The higher crossover rate in the beginning should enable a more extensive exploration of the search space. The smaller crossover rate later on should enable a better exploitation of promising parts. According to (10), the application probability of the macromutations is linearly increased from with the values of p_{MM}^{min} and p_{MM}^{max} being 0.05 and 0.9, respectively. In the early stage of the search the application probability is relatively small since the population is expected to be diverse. One can expect that the population diversity decreases with the number of executed iterations. This is why the application probability of the macromutations is increased, having an ever greater part in the exploration of the search space.

Algorithm 1. DEMM in pseudo-code

```

1: Initialization and parameter setting
2: for  $t := 1 \rightarrow t_{max}$  do
3:   for  $i := 1 \rightarrow NP$  do    %Create trial vector population
4:     if  $\mathcal{U}[0, 1) < p_{MM}$  then
5:       create macromutated vector  $\mathbf{m}_i$  (Eq. (8))
6:        $\mathbf{w}_i := \mathbf{m}_i$ 
7:     else
8:       create mutant/donor vector  $\mathbf{u}_i$  (Eq. (5))
9:       cross over  $\mathbf{v}_i$  and  $\mathbf{u}_i$  to create trial vector  $\mathbf{t}_i$  (Eq. (6))
10:       $\mathbf{w}_i := \mathbf{t}_i$ 
11:    end if
12:  end for
13:  for  $i := 1 \rightarrow NP$  do    %Select new generation
14:    if  $f(\mathbf{w}_i) \leq f(\mathbf{v}_i)$  then
15:       $\mathbf{v}_i := \mathbf{w}_i$ 
16:    end if
17:  end for
18:  update the values of  $p_c$  (Eq. (9)) and  $p_{MM}$  (Eq. (10))
19: end for

```

Table 1. Datasets, used

| Dataset | #instances | #features | #classes |
|-------------------------------|------------|-----------|----------|
| Glass identification | 214 | 9 | 6 |
| Iris plants | 150 | 4 | 3 |
| Yeast | 1484 | 8 | 10 |
| Vowel | 871 | 3 | 6 |
| Ecoli | 336 | 7 | 8 |
| Heart disease | 303 | 13 | 5 |
| Image segmentation | 2310 | 19 | 7 |
| Statlog (Vehicle Silhouettes) | 846 | 18 | 4 |
| Cardiotocography | 2126 | 21 | 10 |

Table 2. Parameter values, used

| Algorithm | Parameter values |
|---------------|---|
| DEMM | $CR = 0.9, F = 0.5$ |
| DE/rand/1/bin | $CR = 0.95, F = 0.5$ |
| DE/best/1/bin | $CR = 0.95, F = 0.9$ |
| DE-RW | $CR = 0.9, F = 0.5$ |
| IDE | $CR = 1 \rightarrow 0.5, F = 0.5 \cdot (1 + \mathcal{U}[0, 1])$ |
| PSO | $\rho_1 = 2, \rho_2 = 2, \omega = 1 \rightarrow 0.7$ |

5 Experimental Analysis

In order to determine the advantages and shortcomings of the proposed algorithm an experimental analysis was conducted on a number of real datasets. The used datasets are described in Table 1. All used datasets were taken from the UCI Machine Learning Repository [17] except the *Vowel*¹ [18] dataset.

5.1 Experiment Setup

The proposed algorithm was compared to four variants of DE and one PSO algorithm. More precisely, it was compared to DE/rand/1/bin, DE/best/1/bin, DE with Random Walk [19] (DE-RW), a DE algorithm with crossover rate and scale factor adjustment as proposed in [20] (IDE), and a PSO algorithm as used in [11] (without the initialization of velocity vectors). The used parameters, obtained in a preliminary analysis, for each algorithm are shown in Table 2. The population size and number of iterations were 50 and 2000, respectively. Meaning, the number of function evaluations was the same for each algorithm. The initial population was generated in the same way for all algorithms; for each representative point a distinct randomly chosen data from the dataset was selected.

¹ <http://www.isical.ac.in/~sushmita/patterns/vowel.dat>

5.2 Results and Discussion

The obtained experimental results are shown in Table 3. They are based on 30 independent runs for each algorithm on each dataset. The table shows the average quality of the found solutions (f_{avg}), their standard deviation (σ), the quality of the best (f_{bst}) and worst (f_{wst}) found solution, and their difference ($f_{wst} - f_{bst}$) as well as the median of the solutions.

According to Table 3, DEMM achieved on average higher quality solutions compared to the other used algorithms. Only on the *Ecoli* dataset did the DE-RW perform insignificantly better than DEMM. On the *Heart* dataset both DEMM and PSO found on average solutions of the same quality, but the standard deviation and the difference between the worst and best solution the PSO algorithm found are significantly bigger. Considering the three highest dimensional datasets used, it can be assumed that DEMM copes better with higher dimensional data compared to the other employed algorithms. All used algorithms performed almost equally well on the *Iris* dataset which is not surprising since it was the simplest dataset used (4 features and 3 classes). However, DE/best/1/bin and PSO, unlike the other employed algorithms, did not find the same solution in every run.

Based on the small standard deviation, difference of the worst and best solution as well as the median of the found solutions, it can be concluded that DEMM is very stable. Therefore, it is reasonable to expect that it will find solutions close (in terms of quality) to the average with a high probability.

Figure 2 shows the quality of the best solution found by each of the algorithms throughout the iterations for the five largest (in terms of the number of instances) datasets used. The figure is based on the average of the same 30 independent runs mentioned earlier. As may be noticed from Fig. 2, both DEMM and DE-RW exhibit similar behavior regarding convergence, but DEMM manages to explore the search space better in the later phases of the search.

By comparing DEMM and DE/rand/1/bin it is possible to comprehend the impact of incorporating the macromutations into the classical DE. The improved performance in terms of solution quality and stability can be mainly attributed to the fact that the macromutations provide a means to explore the search space even when the population converged. Since the macromutations introduce new random vector parameters very large steps within the search space can be made, something which is usually not possible with the common mutation and crossover operators. This can be especially useful in the early phase of the search as a large space can be explored. Also, the macromutations help prevent the algorithm from getting stuck in a local optimum since new vector parameters are introduced independent of the current state of the population.

From the obtained experimental results it can be noted that DE/best/1/bin and PSO performed the worst on average both in terms of solution quality and stability. DE/best/1/bin converges much faster compared to DE/rand/1/bin due to how the base vector is selected, and this is likely the reason why it performed relatively poor (Fig. 2 provides some evidence for that). However, DE/best/1/bin

Table 3. Experimental results

| Dataset | Algorithm | f_{avg} | σ | f_{bst} | f_{wst} | $f_{wst} - f_{bst}$ | median |
|------------|------------|-----------|----------|-----------|-----------|---------------------|--------|
| Glass | DEMM | 0.261 | 0.002 | 0.261 | 0.271 | 0.010 | 0.261 |
| | DE/rand/1/ | 0.266 | 0.006 | 0.261 | 0.277 | 0.017 | 0.263 |
| | DE/best/1/ | 0.283 | 0.035 | 0.261 | 0.420 | 0.160 | 0.275 |
| | DE-RW | 0.262 | 0.003 | 0.261 | 0.275 | 0.015 | 0.263 |
| | IDE | 0.269 | 0.006 | 0.261 | 0.275 | 0.015 | 0.271 |
| | PSO | 0.278 | 0.036 | 0.261 | 0.403 | 0.143 | 0.263 |
| Iris | DEMM | 0.377 | 0.000 | 0.377 | 0.377 | 0.000 | 0.377 |
| | DE/rand/1/ | 0.377 | 0.000 | 0.377 | 0.377 | 0.000 | 0.377 |
| | DE/best/1/ | 0.381 | 0.006 | 0.377 | 0.392 | 0.015 | 0.377 |
| | DE-RW | 0.377 | 0.000 | 0.377 | 0.377 | 0.000 | 0.377 |
| | IDE | 0.377 | 0.000 | 0.377 | 0.377 | 0.000 | 0.377 |
| | PSO | 0.378 | 0.004 | 0.377 | 0.392 | 0.015 | 0.377 |
| Yeast | DEMM | 0.313 | 0.008 | 0.301 | 0.337 | 0.036 | 0.313 |
| | DE/rand/1/ | 0.354 | 0.037 | 0.313 | 0.446 | 0.133 | 0.344 |
| | DE/best/1/ | 0.381 | 0.058 | 0.333 | 0.632 | 0.299 | 0.361 |
| | DE-RW | 0.328 | 0.019 | 0.306 | 0.400 | 0.093 | 0.325 |
| | IDE | 0.387 | 0.034 | 0.343 | 0.491 | 0.148 | 0.378 |
| | PSO | 0.399 | 0.060 | 0.341 | 0.515 | 0.174 | 0.369 |
| Vowel | DEMM | 0.506 | 0.017 | 0.476 | 0.553 | 0.077 | 0.503 |
| | DE/rand/1/ | 0.546 | 0.049 | 0.476 | 0.647 | 0.171 | 0.548 |
| | DE/best/1/ | 0.546 | 0.050 | 0.487 | 0.668 | 0.181 | 0.525 |
| | DE-RW | 0.510 | 0.020 | 0.479 | 0.569 | 0.090 | 0.506 |
| | IDE | 0.507 | 0.029 | 0.476 | 0.587 | 0.111 | 0.500 |
| | PSO | 0.581 | 0.049 | 0.492 | 0.671 | 0.179 | 0.577 |
| Ecoli | DEMM | 0.556 | 0.004 | 0.550 | 0.567 | 0.018 | 0.555 |
| | DE/rand/1/ | 0.572 | 0.014 | 0.549 | 0.609 | 0.061 | 0.568 |
| | DE/best/1/ | 0.587 | 0.016 | 0.561 | 0.623 | 0.062 | 0.585 |
| | DE-RW | 0.555 | 0.006 | 0.549 | 0.570 | 0.021 | 0.553 |
| | IDE | 0.577 | 0.020 | 0.554 | 0.658 | 0.104 | 0.575 |
| | PSO | 0.584 | 0.022 | 0.553 | 0.645 | 0.092 | 0.575 |
| Heart | DEMM | 0.343 | 0.020 | 0.271 | 0.349 | 0.078 | 0.348 |
| | DE/rand/1/ | 0.349 | 0.016 | 0.271 | 0.378 | 0.107 | 0.349 |
| | DE/best/1/ | 0.346 | 0.043 | 0.271 | 0.511 | 0.240 | 0.348 |
| | DE-RW | 0.349 | 0.003 | 0.347 | 0.363 | 0.016 | 0.349 |
| | IDE | 0.346 | 0.015 | 0.271 | 0.363 | 0.092 | 0.349 |
| | PSO | 0.343 | 0.077 | 0.271 | 0.514 | 0.243 | 0.348 |
| Image seg. | DEMM | 0.135 | 0.001 | 0.135 | 0.139 | 0.004 | 0.135 |
| | DE/rand/1/ | 0.529 | 0.167 | 0.135 | 0.781 | 0.646 | 0.535 |
| | DE/best/1/ | 0.561 | 0.138 | 0.281 | 0.769 | 0.489 | 0.561 |
| | DE-RW | 0.137 | 0.004 | 0.135 | 0.143 | 0.008 | 0.135 |
| | IDE | 0.239 | 0.144 | 0.135 | 0.509 | 0.374 | 0.143 |
| | PSO | 0.602 | 0.091 | 0.360 | 0.723 | 0.363 | 0.624 |
| Statlog | DEMM | 0.282 | 0.003 | 0.281 | 0.292 | 0.011 | 0.281 |
| | DE/rand/1/ | 0.339 | 0.010 | 0.321 | 0.345 | 0.023 | 0.345 |
| | DE/best/1/ | 0.339 | 0.011 | 0.321 | 0.363 | 0.042 | 0.345 |
| | DE-RW | 0.310 | 0.020 | 0.281 | 0.345 | 0.064 | 0.321 |
| | IDE | 0.334 | 0.011 | 0.321 | 0.345 | 0.023 | 0.334 |
| | PSO | 0.340 | 0.019 | 0.281 | 0.385 | 0.104 | 0.345 |
| Cardio. | DEMM | 0.502 | 0.042 | 0.446 | 0.576 | 0.131 | 0.490 |
| | DE/rand/1/ | 0.549 | 0.048 | 0.372 | 0.666 | 0.293 | 0.552 |
| | DE/best/1/ | 0.562 | 0.056 | 0.423 | 0.655 | 0.232 | 0.567 |
| | DE-RW | 0.529 | 0.042 | 0.451 | 0.653 | 0.202 | 0.531 |
| | IDE | 0.628 | 0.041 | 0.569 | 0.731 | 0.161 | 0.618 |
| | PSO | 0.600 | 0.061 | 0.526 | 0.815 | 0.288 | 0.574 |

did perform slightly better than the PSO algorithm. The obtained experimental results supports the conclusion reached in [11], on how DE is superior to PSO, at least for data clustering.

Finally, although it is assumed in this paper that the number of clusters is known a priori, the proposed algorithm (DEMM) could be extended in a similar

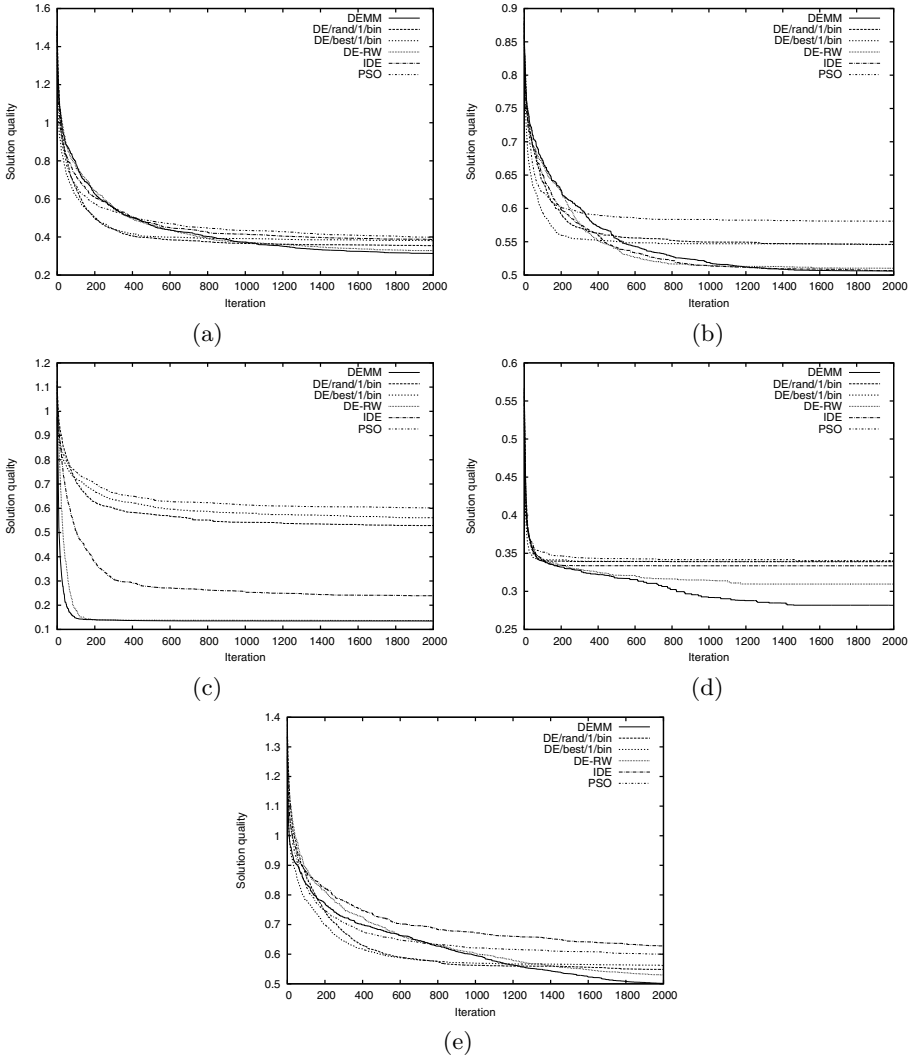


Fig. 2. The quality of the best solution throughout the iterations for datasets (a) Yeast, (b) Vowel, (c) Image segmentation, (d) Statlog (Vehicle Silhouettes), and (e) Cardiocography.

fashion as proposed in [20], in order to provide an automatic (during runtime) determination of the number of clusters inherent to a given dataset. Also, the algorithm could be extended in a way similar to the one presented in [21], as to enable the automatic determination of the proper number of clusters and to perform fuzzy clustering.

6 Conclusion

This paper presented a DE algorithm for clustering. The algorithm incorporates macromutations as an additional mechanism for the exploration of the search space. They are applied with a set probability instead of the usual mutation and crossover. The application probability and macromutation intensity are dynamically adjusted during runtime. This way a more extensive exploration of the search space is enabled in the early phase, and a better exploitation of promising parts in the later phases. The experimental analysis conducted on a number of real datasets showed that the proposed algorithm manages to find high quality solutions and that it is very stable. Future work could include the extension of the algorithm to enable an automatic determination of the proper number of clusters. Also, since in this paper the problem of data clustering was treated as a global optimization problem, the proposed algorithm can be easily applied to other such problems.

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. Jain, A.K.: Data clustering: 50 years beyond K-means. *Pattern Recogn. Lett.* 31, 651–666 (2010)
2. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Comput. Surv.* 31, 264–323 (1999)
3. Murtagh, F., Contreras, P.: Algorithms for hierarchical clustering: an overview. *WIREs Data Mining Knowl. Discov.* 2, 86–97 (2012)
4. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: *5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297 (1967)
5. Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., De Carvalho, A.C.P.L.F.: A survey of evolutionary algorithms for clustering. *IEEE Trans. Sys. Man Cyber. Part C* 39, 133–155 (2009)
6. Laszlo, M., Mukherjee, S.: A genetic algorithm that exchanges neighboring centers for k-means clustering. *Pattern Recogn. Lett.* 28, 2359–2366 (2007)
7. Kwedlo, W.: A clustering method combining differential evolution with the K-means algorithm. *Pattern Recogn. Lett.* 32, 1613–1621 (2011)
8. Hatamlou, A., Abdullah, S., Nezamabadi-pour, H.: A combined approach for clustering based on k-means and gravitational search algorithms. *Swarm Evol. Comput.* 6, 47–52 (2012)
9. Chuang, L.Y., Lin, Y.D., Yang, C.H.: An Improved Particle Swarm Optimization for Data Clustering. In: *International MultiConference of Engineers and Computer Scientists, Hong Kong*, vol. 1 (2012)
10. Zou, W., Zhu, Y., Chen, H., Sui, X.: A Clustering Approach Using Cooperative Artificial Bee Colony Algorithm. *Discret. Dyn. Nat. Soc.*, 16 pages (2010)

11. Paterlini, S., Krink, T.: Differential evolution and particle swarm optimisation in partitionial clustering. *Comput. Stat. Data Anal.* 50, 1220–1247 (2006)
12. Verdramin, L., Campello, R.J.G.B., Hruschka, E.R.: Relative Clustering Validity Criteria: A Comparative Overview. *Stat. Anal. Data Min.* 3, 209–235 (2010)
13. Davies, D.L., Bouldin, D.W.: A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* 1, 224–227 (1979)
14. Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* 11, 342–359 (1997)
15. Das, S., Suganthan, P.N.: Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* 15, 4–31 (2011)
16. Jones, T.: Crossover, Macromutation, and Population-based Search. In: 6th International Conference on Genetic Algorithms, pp. 73–80. Morgan Kaufmann, San Francisco (1995)
17. Bache, K., Lichman, M.: UCI Machine Learning Repository. School of Information and Computer Sciences, University of California, <http://archive.ics.uci.edu/ml>
18. Pal, S.K., Dutta Majumder, D.: Fuzzy sets and decision making approaches in vowel and speaker recognition. *IEEE Trans. Syst., Man, Cybern.* 7, 625–629 (1977)
19. Zhan, Z.-H., Zhang, J.: Enhance Differential Evolution with Random Walk. In: 14th International Conference on Genetic and Evolutionary Computation Conference Companion, pp. 1513–1514. ACM, New York (2012)
20. Das, S., Abraham, A., Konar, A.: Automatic Clustering Using an Improved Differential Evolution Algorithm. *IEEE Trans. Sys. Man Cyber. Part A* 38, 218–237 (2008)
21. Das, S., Konar, A.: Automatic image pixel clustering with an improved differential evolution. *Appl. Soft Comput.* 9, 226–236 (2009)

Improved Adaptive Differential Evolution Algorithm with External Archive

Rammohan Mallipeddi¹ and Ponnuthurai Nagarathnam Suganthan²

¹ Kyungpook National University, Taegu, South Korea, 702 701

² Nanyang Technological University, Singapore, 639 798
mallipeddi@knu.ac.kr, epnsugan@ntu.edu.sg

Abstract. Depending on the complexity of the optimization problem, the performance of differential evolution (DE) algorithm is quite sensitive to the choice of mutation and crossover strategies and their associated control parameters. To obtain optimal performance, while avoiding time consuming parameter tuning, different adaptive and self-adaptive techniques that can update the strategies and/or the parameters during the evolution have been proposed. Adaptive differential evolution with optional archive (JADE) is one of the popular adaptive algorithms that perform well on most of the optimization problems. Motivated by the performance of the JADE algorithm, this paper presents an improved adaptive differential evolution algorithm with external archive (iJADE). Unlike the optional archive in JADE, iJADE algorithm employs an external archive which is updated every generation by tournament selection to incorporate the parents which cannot progress to the next generation. In addition, iJADE uses an ensemble of two crossover strategies, binomial and exponential, instead of a single crossover strategy as in JADE. The performance of the algorithm is evaluated on a set of 16 bound-constrained problems designed for Conference on Evolutionary Computation (CEC) 2005 and is compared with JADE algorithm.

Keywords: Differential Evolution, Global optimization, Parameter adaptation, External archive.

1 Introduction

Differential evolution (DE) [1], a population based stochastic search technique, has been successfully applied in diverse areas such as mechanical engineering [2], communication [3], optics [4], pattern recognition [5], signal processing [6] and power systems [7]. However, it has been demonstrated, experimentally [8, 9] and theoretically [10], that the performance of DE is sensitive to the mutation strategy, crossover strategy and control parameters such as population size (NP), crossover rate (CR) and scale factor (F). The best combination of strategies and their associated control parameters can be different for different optimization problems. In addition, for the same optimization problem the best combination can vary depending on the available computational resources and accuracy requirements [11]. Therefore, to successfully solve

a specific optimization problem, it is necessary to perform trial-and-error search for the most appropriate combination of strategies and their associated parameter values. However, the trial-and-error search process is time-consuming and incurs high computational costs. Therefore, to overcome the time consuming trial-and-error procedure different adaptation schemes [12-16] have been proposed in the literature. In addition, motivated by the observation that the population of DE may evolve through different regions in the search space, within which different strategies with different parameter settings may be more effective than others, the authors in [11] proposed a DE algorithm based on the idea of ensemble strategies and parameters.

Among the different adaptive DE variants, adaptive differential evolution with optional external archive (JADE) is good in terms of convergence speed and robustness on a variety of optimization problems [17]. In this paper, we propose an improved JADE algorithm (iJADE) which employs an ensemble of different crossover strategies and a compulsory archive that is updated using tournament selection.

The remainder of this paper is organized as follows: Section 2 presents a literature survey on different adaptive DE variants. Section 3 presents the proposed iJADE algorithm. Section 4 presents the experimental results and discussions while Section 5 concludes the paper.

2 Literature Review

Differential Evolution (DE) is a real-coded global optimization algorithm over continuous spaces [18]. As complexity of the optimization problem increases the performance of DE algorithm becomes more sensitive to the strategy and the associated parameter values [8]. Therefore, inappropriate choice of mutation and crossover strategies and their associated parameters may lead to premature convergence, stagnation or wastage of computational resources [8, 15, 19-21]. In literature, various empirical guidelines were suggested for choosing the appropriate strategies and control parameter settings depending on the characteristics of the optimization problems [8, 18, 22-24]. However, depending on the complexity of the optimization problem, choosing an appropriate mutation strategy and control parameters is not straight forward due to the complex interaction of control parameters with the DE's performance [12]. In addition, the manual setting and/or tuning of DE strategies and parameters based on the guidelines result in various conflicting conclusions, which lack sufficient justifications. Therefore, to avoid the tuning of parameters by trial-and-error procedure, various adaptive techniques have been proposed [15, 19, 25-27].

In [26], an adaptive DE algorithm that adapts the control parameters F and CR based on fuzzy logic controllers referred to as FADE was proposed. In FADE, the inputs to the fuzzy controller are the relative function values of individuals in the successive generations. In [15], DE parameter adaptation based on controlling the population diversity and multi-population approach (ADE) was proposed, which was later extended to form an adaptive Pareto DE algorithm for multi-objective optimization [27]. In [25], the authors propose a DE algorithm for multi-objective optimization problems, where the crossover rate parameter is self-adapted or simultaneously evolved with other parameters by encoding into each individual. Unlike the crossover rate, the scale factor F was generated using a Gaussian distribution $N(0,1)$ for each

individual. In [14], Qin *et al.* proposed a self-adaptive DE algorithm (SaDE) in which the mutation strategies and the respective control parameter are self-adapted based on their previous experiences of generating promising solutions. In SaDE, the scale factor (F) was randomly generated with a mean and standard deviation of 0.5 and 0.3, respectively. In [13], a self-adaptation scheme (SDE) in which CR is generated randomly for each individual using a normal distribution $N(0.5,0.15)$, while scale factor F is adapted analogous to the adaptation of crossover rate CR in [25] was introduced. In [12], the authors proposed a self-adaptation scheme (JDE) in which control parameters F and CR are encoded into the individuals and are adjusted by introducing two new parameters τ_1 and τ_2 . Initially, the parameters F and CR of each individual were assigned to 0.5 and 0.9 respectively and are adjusted (during every generation of the evolution by generating a uniform random number $rand$ in the range of $[0, 1]$. If $rand < \tau_1$, the parameter F was reinitialized to a new random value in the range $[0.1, 1.0]$. Otherwise, it was kept unchanged. Similarly, if, $rand < \tau_2$ then parameter CR was reinitialized in the range $[0, 1]$. Otherwise, it is kept unchanged.

3 Improved Adaptive Differential Evolution with External Archive

3.1 Adaptive DE Algorithm with Optional Archive (JADE)

Adaptive differential evolution algorithm with optional archive (JADE) [17] implements a mutation strategy “DE/current-to- p best” as a generalization to the classic “DE/current-to-best” strategy. Unlike the classic mutation strategy which uses the current best individual, “DE/current-to- p best” utilizes the information present in p fitter individuals of the current population. The use of multiple solutions helps in balancing the greediness of the mutation and the diversity of the population. Parent individual solutions of the current generation which fail to make to the next generation of the evolution process during the selection process are stored in the optional archive. In other words, the set of recently explored *inferior* solutions stored in the archive provide the historical information regarding the progress direction of the search. In JADE, the control parameters (F and CR) are updated in an adaptive manner in order to alleviate the trial and error search. Hence, it is not necessary that the users have a prior knowledge of the relationship between the parameter settings and the characteristics of optimization problems. In JADE, the optional archive and the parameter updating diversify the population to provide robustness and improve the convergence performance of the algorithm.

In JADE, using the DE/current-to- p best with archive, a mutation vector corresponding to the individual X_i in generation G is generated as:

$$V_{i,G} = X_{i,G} + F_i(X_{best,G}^p - X_{i,G}) + F_i(X_{r1,G} - \bar{X}_{r2,G}) \quad (1)$$

where $X_{r1,G}$ and $X_{best,G}^p$ are selected from the current population (\mathbf{P}), while $\bar{X}_{r2,G}$ is randomly chosen from the union of current population (\mathbf{P}) and the archive (\mathbf{A}). As mentioned earlier, the recently explored inferior solutions in archive (\mathbf{A}) in addition to the current population provide the additional information regarding the promising progress direction and are also capable of improving the diversity of the population.

In JADE, the archive is initially empty and after every generation the parent solutions that fail in the selection process are added to the archive. When the size of the archive exceeds a certain threshold (NP_{arch}), some of the solutions are randomly removed from the archive to maintain the archive size constant at NP_{arch} .

At each generation, the scale factor F_i and crossover probability CR_i of each individual X_i is independently generated as

$$F_i = \text{randc}_i(\mu_F, 0.1) \quad (2)$$

$$CR_i = \text{randn}_i(\mu_{CR}, 0.1) \quad (3)$$

As shown in eqns. (2) and (3), the parameters F and CR corresponding to each individual are sampled using Cauchy and Normal distributions, respectively. Then mean values μ_F and μ_{CR} are initialized to 0.5 and are updated at the end of each generation as

$$\mu_F = (1 - c) \cdot \mu_F + c \cdot \text{mean}_L(S_F) \quad (4)$$

$$\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot \text{mean}_A(S_{CR}) \quad (5)$$

where c is a positive constant between 0 and 1. The terms $\text{mean}_A(\cdot)$ and $\text{mean}_L(\cdot)$ denote the arithmetic mean and Lehmer mean [17], respectively. S_F and S_{CR} denote the sets of mutation factors and crossover probabilities, respectively that produced successful trial vectors in the previous generation.

3.2 Improved Adaptive DE Algorithm with External Archive (iJADE)

JADE algorithm employs a mutation strategy that balances the greediness and robustness. However, the performance of DE algorithm also depends on the crossover strategy used. According to the literature [28], binomial and exponential crossover strategies that are commonly employed have their own advantages and disadvantages in solving the optimization problems with different characteristics. Binomial crossover is good at handling un-linked or un-rotated problems, while exponential crossover is good at handling linked or rotated problems. JADE algorithm uses the binomial crossover. According to the authors in [11], the combination or ensemble of the two crossover strategies of DE can be better than using a single strategy.

In JADE, the optional archive provides information regarding the progressive search direction and improves the robustness by increasing the diversity of the population. In JADE, the authors use a simple strategy to update the existing archive. The inferior solutions of the current generation are added to the archive and if the archive exceeds a predefined size the members of the archive are removed randomly. Since the external archive provides information regarding the search direction, maintaining the fitter individuals in the archive would be better as they provide more useful information compared to the less fit individuals. Therefore, removing the archive members in a random manner may not be a better way since some the useful information may be lost due to the removal of fitter solutions. Hence, a better updating of the archive that can preserve the fitter individuals can improve the performance of the algorithm.

The proposed iJADE algorithm is a modification of the JADE algorithm, which employs an ensemble of the crossover strategies and updates the archive by using tournament selection. The two modifications are described below and the outline of iJADE is shown in Table 1.

Table 1. Improved Adaptive Differential Evolution with External Archive (iJADE)

1. Randomly initialize a population $\mathbf{P}_G = \{X_{i,G}, \dots, X_{NP,G}\}$ with $X_{i,G} = \{x_{i,G}^1, \dots, x_{i,G}^D\}$, $i = 1, \dots, NP$ uniformly distributed in the range $[X_{\min}, X_{\max}]$, where $X_{\min} = \{x_{\min}^1, \dots, x_{\min}^D\}$ and $X_{\max} = \{x_{\max}^1, \dots, x_{\max}^D\}$. Set $\mu_{CRbio} = 0.5$, $\mu_{Fbio} = 0.5$, $\mu_{Fexp} = 0.5$, $A = \phi$.
2. With equal probability, individuals of current population \mathbf{P} are assigned to one of \mathbf{P}_{bio} and \mathbf{P}_{exp} depending on the crossover strategy to be used to generate the trial vector
3. **WHILE** stopping criterion is not satisfied
 - $SF_{bio} = \Phi$; $SF_{exp} = \Phi$; $SCR_{bio} = \Phi$;
 - FOR** $i = 1: NP$
 - Generate CR_i and F_i using eqns. (2) and (3)
 - Generate the mutant vector ($V_{i,G}$) using the equation (1)
 - IF** $X_i \in P_{bio}$
 - Generate the trial vector ($U_{i,G}$) using Binomial Crossover [28]
 - ELSE IF** $X_i \in P_{exp}$
 - Generate the trial vector ($U_{i,G}$) using Exponential Crossover [28]
 - END IF**
 - IF** $f(U_{i,G}) \leq f(X_{i,G})$
 - $X_{i,G+1} = U_{i,G}$, $X_{i,G} \rightarrow A$
 - IF** $X_i \in P_{bio}$
 - $CR_i \rightarrow SCR_{bio}$, $F_i \rightarrow SF_{bio}$
 - ELSE**
 - $F_i \rightarrow SF_{exp}$
 - END IF**
 - ELSE**
 - $X_{i,G+1} = X_{i,G}$
 - END IF**
 - END FOR**
 - Update archive A so that $|A| \leq NP_{arch}$
 - Update μ_F and μ_{CR} using eqns. (4) and (5)
 - $G = G + 1$
 - END WHILE**

3.2.1 Ensemble of Crossover Strategies

In iJADE, the members of the initial population (\mathbf{P}) are divided in to two groups \mathbf{P}_{bio} and \mathbf{P}_{exp} . Members belonging to \mathbf{P}_{bio} generate trial vectors from the mutant vector using binomial crossover while the members belonging to \mathbf{P}_{exp} generate trial vectors using exponential crossover. The generation and updating of the scale factor F for the individuals of \mathbf{P}_{bio} and \mathbf{P}_{exp} done separately using eqns. (2) and (4). The crossover

probability corresponding to the binomial crossover is done using the eqns. (3) and (5). However, in iJADE the crossover probability of the exponential crossover is fixed to be constant at 0.9 and is not adapted unlike the crossover probability of the binomial crossover [11].

3.2.2 Updating the External Archive

Unlike in JADE, after combining the inferior solutions of the current generation with solutions in the archive, if the archive size exceeds the predefined size (NP_{arch}) then the members of the archive participate in a tournament selection. The members of the archive with maximum number of wins are retained while the individuals with least number of the wins are discarded. The selection of the archive individuals based on the tournament selection helps in retaining the fitter individuals among the solutions that are unable to make into the population. The fitter individuals in the archive provide better information regarding the search direction compared to the randomly selected individuals.

4 Experimental Results

To evaluate the performance of the algorithms the 16 bound-constrained test problems of CEC 2005 are used. The maximum numbers of function evaluations used are

Table 2. Results for 10D benchmark problems of CEC 2005

| Fcn | 10D | | | | | |
|----------|------------------|----------|----------------|----------|-----------------|-----------------|
| | JADE w/o Archive | | JADE w Archive | | iJADE | |
| | Mean | Std | Mean | Std | Mean | Std |
| f_1 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_2 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_3 | 1.11E-25 | 4.03E-26 | 4.18E-26 | 9.50E-27 | 1.24E-26 | 2.03E-26 |
| f_4 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_5 | 2.42E-13 | 6.28E-13 | 5.46E-13 | 8.79E-13 | 4.24E-13 | 7.82E-13 |
| f_6 | 8.13E-01 | 1.67E+00 | 1.92E+00 | 2.09E+00 | 0 | 0 |
| f_7 | 7.51E-03 | 6.61E-03 | 1.10E-02 | 1.11E-02 | 7.63E-03 | 6.61E-03 |
| f_8 | 2.03E+01 | 1.54E-01 | 2.03E+01 | 3.34E-02 | 2.00E+01 | 1.54E-01 |
| f_9 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{10} | 4.39E+00 | 1.09E+00 | 4.39E+00 | 1.49E+00 | 3.39E+00 | 1.42E+00 |
| f_{11} | 4.41E+00 | 1.03E+00 | 4.87E+00 | 5.91E-01 | 1.40E+00 | 1.47E+00 |
| f_{12} | 8.89E+01 | 2.89E+02 | 9.26E+00 | 1.09E+01 | 6.37E+00 | 7.81E+00 |
| f_{13} | 2.48E-01 | 5.43E-02 | 2.77E-01 | 4.15E-02 | 2.51E-01 | 6.41E-02 |
| f_{14} | 2.74E+00 | 3.01E-01 | 2.79E+00 | 3.74E-01 | 2.44E+00 | 4.81E-01 |
| f_{15} | 8.51E+01 | 1.49E+02 | 1.37E+02 | 1.77E+02 | 5.88E+01 | 1.24E+02 |
| f_{16} | 9.77E+01 | 4.41E+00 | 9.50E+01 | 4.08E+00 | 9.13E+01 | 4.37E+00 |

Table 3. Results for 30D benchmark problems of CEC 2005

| Fcn | 30D | | | | | |
|----------|------------------|----------|----------------|----------|-----------------|-----------------|
| | JADE w/o Archive | | JADE w Archive | | iJADE | |
| | Mean | Std | Mean | Std | Mean | Std |
| f_1 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_2 | 8.59E-28 | 4.19E-28 | 4.05E-28 | 2.19E-28 | 3.15E-28 | 4.32E-28 |
| f_3 | 7.96E+03 | 3.88E+03 | 5.28E+03 | 4.09E+03 | 3.94E+03 | 2.36E+03 |
| f_4 | 2.45E-02 | 8.40E-02 | 6.21E-08 | 9.64E-08 | 8.72E-23 | 4.62E-22 |
| f_5 | 7.53E+02 | 3.68E+02 | 5.15E+02 | 6.47E+02 | 4.54E+02 | 1.81E+02 |
| f_6 | 1.03E+01 | 2.72E+01 | 2.79E+01 | 7.55E+00 | 6.62E-01 | 1.51E+00 |
| f_7 | 1.56E-02 | 1.53E-02 | 6.91E-03 | 6.12E-03 | 7.45E-03 | 3.56E-03 |
| f_8 | 2.08E+01 | 2.46E-01 | 2.09E+01 | 5.50E-02 | 2.08E+01 | 2.25E-01 |
| f_9 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{10} | 2.73E+01 | 5.70E+00 | 2.43E+01 | 6.23E+00 | 2.45E+01 | 7.37E+00 |
| f_{11} | 2.68E+01 | 2.03E+00 | 2.64E+01 | 1.93E+00 | 2.28E+01 | 3.53E+00 |
| f_{12} | 4.92E+03 | 3.97E+03 | 4.45E+03 | 4.53E+03 | 2.71E+03 | 3.77E+03 |
| f_{13} | 1.67E+00 | 3.05E-02 | 1.14E+00 | 1.27E-01 | 1.07E+00 | 1.82E-01 |
| f_{14} | 1.24E+01 | 3.27E-01 | 1.24E+01 | 2.15E-01 | 1.23E+01 | 2.57E-01 |
| f_{15} | 3.20E+02 | 1.17E+02 | 3.50E+02 | 1.08E+02 | 2.70E+02 | 1.05E+02 |
| f_{16} | 1.45E+01 | 1.55E+01 | 6.22E+01 | 2.06E+01 | 5.48E+01 | 2.81E+01 |

100000 and 300000 for 10D and 30D problems respectively. The parameters are set as: $NP = 50$ and $NP_{arch} = 1.5 \times NP$. The remaining parameters are maintained the same as in the original JADE algorithm. The experimental results for 10D and 30D benchmark problems are presented in Tables 2 and 3, respectively. In Tables 2 and 3, for a particular problem, the results are highlighted or mentioned to be significant if the performance of the algorithm is statistically significant. The statistical significance is decided by performing a statistical t -test with a significance level of 0.05.

On 10D problems, comparing the performance of JADE with and without archive shows that JADE with archive shows significant improvement on problems f_{12} and f_{16} while JADE without archive performs well on f_6 , f_7 and f_{15} . Comparing the performance of JADE with and without archive, on 30D problems, show that JADE with archive shows significant improvement on problems f_4 , f_7 , f_{10} , f_{12} and f_{16} while JADE without archive performs well on f_6 , and f_{15} . Therefore, as observed in [17], JADE with archive shows better improvements on high dimensional problems than on low dimensional problems. On some of the low dimensional problems the performance of JADE without archive is better than the JADE with archive.

In Tables 2 and 3, to compare the performance of iJADE and JADE algorithms the results of the algorithm with significant performance are highlighted. From the highlighted results it can be observed that iJADE is shows better or similar performance over all the benchmark problems and is never worse. On 10D problems, it can be observed that iJADE shows significant improvement on problems f_{10} , f_{11} , f_{12} , f_{14} , f_{15}

and f_{16} . On 30D problems, iJADE shows superior performance on $f_3, f_4, f_5, f_6, f_{11}, f_{12}, f_{14}, f_{15}$ and f_{16} . Unlike the JADE with archive, the iJADE algorithm is never worse than the JADE without archive. The improved performance can be attributed to the new updating scheme of the archive. The superior performance of iJADE on 30D versions of f_3 and f_4 can be attributed to the ensemble of crossover strategies.

5 Conclusions

Adaptive differential evolution with optional archive (JADE) is one of the most popular adaptive DE variants. In this paper, we proposed an improved version of JADE referred to as iJADE. Unlike JADE, iJADE updates the archive using a tournament selection which helps to preserve the fitter individuals in the archive. The fitter individuals in the archive provide more useful information regarding the search direction. In addition, iJADE employs an ensemble of binomial and exponential crossover strategies unlike the single binomial crossover in JADE. On a given set of benchmark problems, the ensemble of crossover strategies provides robustness compared to a single crossover strategy. The performance of iJADE is evaluated on set of 16 benchmark problems and is favorably compared with the JADE algorithm with optional archive.

Acknowledgment. This research was supported by Kyungpook National University Research Fund, 2012 (201227350000).

References

1. Storn, R., Price, K.: Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces. Technical Report TR-95-012, ICSI (1995), <http://http://http://icsi.berkeley.edu/~storn/litera.html>
2. Joshi, R., Sanderson, A.C.: Minimal representation multisensor fusion using differential evolution. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans* 29(1), 63–76 (1999)
3. Mallipeddi, R., et al.: Robust Adaptive Beamforming Based on Covariance Matrix Reconstruction for Look Direction Mismatch. *Progress in Electromagnetic Research Letters* 25, 37–46 (2011)
4. Venu, M.K., Mallipeddi, R., Suganthan, P.N.: Fiber Bragg grating sensor array interrogation using differential evolution. *Optoelectronics and Advanced Materials - Rapid Communications* 2(11), 682–685 (2008)
5. Das, S., Konar, A.: Automatic image pixel clustering with an improved differential evolution. *Applied Soft Computing* 9(1), 226–236 (2009)
6. Storn, R.: Differential evolution design of an IIR-filter. In: Storn, R. (ed.) *IEEE International Conference on Evolutionary Computation 1996*, pp. 268–273. IEEE (1996)
7. Mallipeddi, R., et al.: Efficient constraint handling for optimal reactive power dispatch problems. *Swarm and Evolutionary Computation* 5, 28–36 (2012)
8. Gämperle, R., Müller, S.D., Koumoutsakos, P.: A Parameter Study for Differential Evolution. In: *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation 2002*, pp. 293–298. WSEAS Press, Interlaken (2002)
9. Liu, J., Lampinen, J.: On setting the control parameter of the differential evolution method. In: *Proc. 8th Int. Conf. Soft Computing, MENDEL 2002* (2002)

10. Jingqiao, Z., Sanderson, A.C.: An approximate gaussian model of Differential Evolution with spherical fitness functions. In: IEEE Congress on Evolutionary Computation, CEC 2007 (2007)
11. Mallipeddi, R., et al.: Differential Evolution Algorithm with Ensemble of Parameters and Mutation Strategies. *Applied Soft Computing* 11(2), 1679–1696 (2011)
12. Brest, J., et al.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation* 10(8), 646–657 (2006)
13. Omran, M.G.H., Salman, A., Engelbrecht, A.P.: Self-adaptive Differential Evolution. In: Hao, Y., Liu, J., Wang, Y.-P., Cheung, Y.-m., Yin, H., Jiao, L., Ma, J., Jiao, Y.-C. (eds.) CIS 2005. LNCS (LNAI), vol. 3801, pp. 192–199. Springer, Heidelberg (2005)
14. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation* 13(2), 398–417 (2009)
15. Zaharie, D.: Control of Population Diversity and Adaptation in Differential Evolution Algorithms. In: Proceedings of the 9th International Conference on Soft Computing, Brno, pp. 41–46 (2003)
16. Tvrdik, J.: Adaptation in differential evolution: A numerical comparison. *Applied Soft Computing* 9(3), 1149–1155 (2009)
17. Zhang, J.: JADE: Adaptive Differential Evolution with Optional External Archive. *IEEE Transactions on Evolutionary Computation* 13(5), 945–958 (2009)
18. 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)
19. Das, S., Konar, A., Chakraborty, U.K.: Two Improved Differential Evolution Schemes for Faster Global Search. In: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 991–998 (2005)
20. Lampinen, J., Zelinka, I.: On Stagnation of the Differential Evolution Algorithm. In: Proceedings of MENDEL 2000, 6th International Mendel Conference on Soft Computing, pp. 76–83 (2000)
21. Price, K.V., Storn, R.M., Lampinen, J.A. (eds.): *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series. Springer, Berlin (2005)
22. Storn, R., Price, K.: Differential Evolution: A Simple Evolution Strategy for Fast Optimization. *Dr. Dobb's Journal* 22(4), 18–24 (1997)
23. Storn, R.: On the Usage of Differential Evolution for Function Optimization. In: Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS), pp. 519–523. IEEE, Berkeley (1996)
24. Rönkkönen, J., Kukkonen, S., Price, K.V.: Real-parameter optimization with differential evolution. In: IEEE Congress on Evolutionary Computation (2005)
25. Abbass, H.A.: The Self-Adaptive Pareto Differential Evolution Algorithm. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 1, pp. 831–836 (2002)
26. Liu, J., Lampinen, J.: A fuzzy adaptive differential evolution algorithm. *Soft Computing* 9(6), 448–462 (2005)
27. Zaharie, D., Petcu, D.: Adaptive Pareto differential evolution and its parallelization. In: Proc. of 5th International Conference on Parallel Processing and Applied Mathematics, Czestochowa, Poland (2003)
28. 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)

Fuzzy Clustering of Image Pixels with a Fitness-Based Adaptive Differential Evolution

Soham Sarkar^{1,*}, Gyana Ranjan Patra²,
Swagatam Das³, and Sheli Sinha Chaudhuri⁴

¹ Department of Electronics and Communication Engineering, RCC Institute of Information Technology, Kolkata, 700015, India

² Department of Electronics and Communication Engineering, Institute of Technical Education & Research, S'o'A University, Bhubaneswar, 751030, India

³ Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, 700108, India

⁴ Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata, 700032, India

sarkar.soham@gmail.com, gyana.patra@soauniversity.ac.in,
swagatamdas19@yahoo.co.in, shelism@rediffmail.com

Abstract. Fuzzy c-means is currently the most widely used pixel clustering technique in the field of automatic image segmentation. However performance of fuzzy c-means largely relies upon the selection of initial cluster centers. For improper choice of initial cluster centers, FCM sometimes gets stuck at local optima. In this work a modified Fitness-Based Adaptive Differential Evolution algorithm is presented for clustering the pixels of an image. The control parameters, which are crucial for convergence of Differential Evolution, are chosen adaptively based on fitness statistics of population. For performance measurements, the Berkley Segmentation Dataset and Benchmarks (BSD 300) is used. The outcomes of the proposed algorithm are compared with the famous fuzzy c-means algorithm both qualitatively and quantitatively.

Keywords: Differential Evolution (DE), Berkley Segmentation Dataset and Benchmarks (BSD 300), Adaptive, Unsupervised, Fuzzy clustering, image segmentation, L^*a*b^* color space.

1 Introduction

IMAGE segmentation can be viewed as the process of dividing an image into disjoint homogeneous regions which contain similar objects of interest or part of them. Among the several image segmentation techniques, image pixel clustering is perhaps the most popular one. Clustering can be defined as the optimal partitioning of a given set of N data points into K subgroups, such that data points belonging to the same group are as similar to each other as possible whereas data points from two different groups share the maximum difference [1].

* Corresponding author.

Image clustering techniques can be broadly categorized into two classes: supervised and unsupervised. Automatic clustering or partitioning of unlabeled feature vectors obtained from an image is however the most challenging one. The resultant regions obtained from clustering can then be labeled by hand or by some automated process. Over the years many clustering algorithms have been applied to solve the multi-class segmentation problem (e.g., k-means [2], fuzzy c-means (FCM) [3], ISODATA [4] and Snob [5]). The FCM [6] seems to be the most popular candidate in the field of clustering algorithms.

The biggest challenge associated with FCM is the initialization of cluster centers. It often gets stuck in local optima instead of global optima if it gets started with poor initialization. Several solutions have been proposed in the literature to overcome this problem, amongst them the ones that incorporate the usage of metaheuristics seem to be the most promising ones. These algorithms formulate the clustering method of FCM as a global optimization problem. Researchers suggested several global optimization techniques like, Tabu search [7], Simulated Annealing [7, 9], Genetic Algorithm [10, 11] to improve the performance of fuzzy clustering. Differential Evolution, probably the most powerful global optimizer of recent time [12, 13] is applied after fuzzy clustering to obtain the global optima so that it leads to more accurate image segmentation. The performance of DE dependent majorly on its control parameter scale factor F and cross over rate Cr . Several researchers tried different modification of DEs and used them in pixel clustering problems [14, 15]. In this paper we illustrate a competent technique for image segmentation using DEs by varying F , Cr in accordance with fitness value of the population, proposed by Patra et. al. [16].

The results are computed over 300 color images of The Berkley segmentation Dataset and Benchmarks (BSD 300). Also we have used $L^*a^*b^*$ color space instead of RGB color space. The results are compared with FCM both qualitatively and quantitatively.

The rest of the paper is composed as following. The basic concept Fuzzy clustering problem is described in Section 2. A brief introduction of Differential Evolution (DE) is given in Section 3. In Section 4 detailed discussion is given on Fitness-based Adaptive Differential Evolution (FBADe). The experimental results and comparative performances are presented in Section 4. Lastly the paper is concluded in Section 5.

2 Classical Fuzzy C-Means Algorithm

In the classical FCM algorithm the objective cluster center is obtained by minimizing a *within cluster sum* function J_m , which can be defined by [3]

$$J_m = \sum_{j=1}^N \sum_{i=1}^K (u_{ij})^m \|\vec{X}_j - \vec{V}_i\|^2 \quad (1)$$

Where, K denotes number of cluster center for N data points, u_{ij} denotes fuzzy membership of j^{th} point in the i^{th} cluster, m is the degree of fuzziness, $\|\cdot\|$

denotes the distance between i^{th} cluster center \vec{V}_i and j^{th} d -dimensional dataset \vec{X}_j . At start of the FCM, the cluster K centers are needed to be initialized randomly and then in each iteration, the fuzzy membership of each data point is calculated to determine their belongingness to \vec{V}_i cluster center can be determined using

$$u_{ij} = \frac{\|\vec{X}_j - \vec{V}_i\|^{\frac{2}{1-m}}}{\sum_{k=1}^K \|\vec{X}_j - \vec{V}_k\|^{\frac{2}{1-m}}} \tag{2}$$

The value of m is greater than 1. The K cluster centers can be determined by following equation

$$\vec{V}_i = \frac{\sum_{j=1}^N (u_{ij})^m \vec{X}_j}{\sum_{j=1}^N (u_{ij})^m} \tag{3}$$

The algorithm terminates if the cluster centers are not updated further. Finally, each data point is assigned to the cluster to which it has maximum membership. However FCM algorithm sometimes gets stuck at some suboptimal solution due to improper initialization. DE is used to overcome this problem.

3 Differential Evolution (DE)

DE, a population-based global optimization algorithm, was proposed by Storn in 1997. The i^{th} individual (parameter vector) of the population at generation (time) t is a D -dimensional vector containing a set of D optimization parameters:

$$\vec{Z}_i(t) = [Z_{i,1}(t), Z_{i,2}(t), \dots, Z_{i,D}(t)] \tag{4}$$

In each generation to change the population members $\vec{Z}_i(t)$ (say), a *donor* vector $\vec{Y}_i(t)$ 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,

$$\vec{Y}_{i,j}(t) = Z_{r_1,j}(t) + F \cdot (Z_{r_2,j}(t) - Z_{r_3,j}(t)) \tag{5}$$

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 $\overline{Z}_i(t)$, a trial vector $\overline{R}_i(t)$ is created in the following fashion:

$$\begin{aligned} R_{i,j}(t) &= Y_{i,j}(t) && \text{if } \text{rand}_j(0,1) \leq Cr \text{ or } j = \text{rn}(i) \\ &= Z_{i,j}(t) && \text{if } \text{rand}_j(0,1) > Cr \text{ or } j \neq \text{rn}(i) \end{aligned} \quad (6)$$

For $j = 1, 2, \dots, D$ and $\text{rand}_j(0,1) \in [0,1]$ is the j^{th} evaluation of a uniform random number generator. $\text{rn}(i) \in [1, 2, \dots, D]$ is a randomly chosen index to ensure that $\overline{R}_i(t)$ gets at least one component from $\overline{Z}_i(t)$. 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:

$$\begin{aligned} \overline{Z}_i(t+1) &= \overline{R}_i(t) && \text{if } f(\overline{R}_i(t)) \leq f(\overline{Z}_i(t)) \\ &= \overline{Z}_i(t) && \text{if } f(\overline{R}_i(t)) > f(\overline{Z}_i(t)) \end{aligned} \quad (7)$$

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

4 Fitness-Based Adaptive Differential Evolution (FBADE)

In this paper we illustrate a competent technique for image segmentation using DEs. The two DE parameters viz., scale factor F , and crossover rate Cr , are known to critically affect the behavior and performance of the optimization process. These generalized guidelines are inadequate as the choice of the optimal F , and Cr , becomes specific to the problem under consideration. In these situations the user has to set a number of behavioural parameters that influence the performance of this process, see for example Storn et al. [12], Liu and Lampinen [17], and Zaharie [18].

There has been a trend in recent years to try and make the DE parameters automatically adapt to new problems during optimization, hence alleviating the need for the practitioner to select the parameters by hand, see for example Price et al. [19], Liu and Lampinen [20], Qin et al. [21-22], Brest et al. [23], Mallipeddi et al. [24], Islam et al. [25].

We have chosen to use adaptive values of F , and Cr for improving the convergence capacity of the DE. The adaptation process used here is dependent on the fitness statistics of the population. With such an approach we aim to solve the problem of choosing the optimal values F , and Cr for the DE. This version, called fitness-based adaptive differential evolution (FBADE), was proposed for data clustering [16]. The main idea behind this adaptation mechanism is that the search-agents (DE-vectors) placed near to the optimum have small mutation step-size and during crossover, it passes more genetic information to its offspring for better exploitation.

The proposers of DE suggested that control parameters should be adjusted as $F \in [0.5, 1]$, and $Cr \in [0.8, 1]$ [26]. We are going to use an adaptation process based on the fitness of the population pool which was originally proposed for the control of parameters of GA [27]. The *DE/rand/1/bin* scheme was used with each vector being extended with its own F and Cr values. In this process control parameters were adjusted in every generation for each individual according to the scheme shown in Eqn. (8)

$$\begin{aligned}
 F_{i,G+1} &= \begin{cases} k_1 & \text{if } f_{i,G} \leq \bar{f}_G \\ k_2 \frac{f_{max,G+1} - f_{i,G}}{f_{max,G} - \bar{f}_{i,G}} & \text{if } f_{i,G} > \bar{f}_G \end{cases} \\
 Cr_{i,G+1} &= \begin{cases} k_3 & \text{if } f_{i,G} \leq \bar{f}_G \\ k_4 \frac{f_{max,G+1} - f_{i,G}}{f_{max,G} - \bar{f}_{i,G}} & \text{if } f_{i,G} > \bar{f}_G \end{cases}
 \end{aligned} \tag{8}$$

Where $f_{max,G+1}$ and $f_{i,G}$ are the maximum fitness values in the $(G + 1)^{th}$ and G^{th} generations respectively. $f_{i,G}$ is the fitness of the i^{th} individual of G^{th} generation, \bar{f}_G is the average fitness of the generation G . $F_{i,G+1}$ and $Cr_{i,G+1}$ are the values of scale factor and crossover rate of the individual in the generation G . The values of k_1, k_2, k_3 and k_4 are chosen to be in the range $[0, 1]$.

5 Experimental Results

The simulations are performed with MATLAB R2012a in Intel® Core™ i3 3.2 GHz processor based workstation. For testing and analysis, 300 color images are used from the Berkeley Segmentation Dataset and Benchmark (BSDS 300) [28]. For statistical comparison purposes four of the performance evaluation metrics developed by the Berkeley image segmentation database, are used. The matrices are: Probabilistic Rand Index (PRI), Variation of Information (VoI), Global Consistency Error (GCE) and Boundary Displacement Error (BDE). *Higher* value of PRI indicates better segmentation, whereas for the rest of the metrics, lower values point out the same [28]. The RGB images of BSD300 are converted to $L^*a^*b^*$ color space (CIE 1976) and L^* component is used for clustering as it is closed to human perception of lightness. The components of $L^*a^*b^*$ color space along with original RGB color image is shown in Fig. 1. For better understanding, the outcome images are displayed in average color image. No of clusters are maintained as $K = 5$. The detailed experimental setup of FBADE is given in Table 1. Here the dimension D equals K .

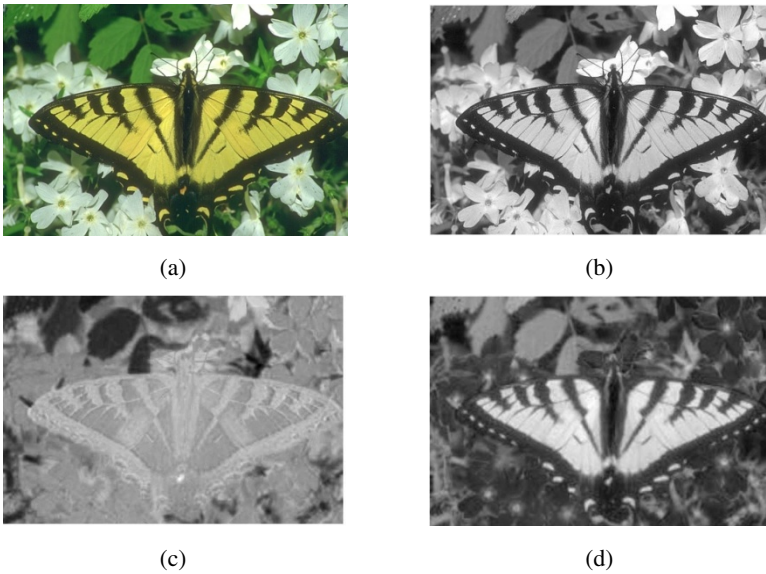


Fig. 1. (a) Original color Image, (b)–(d) Three components of (a) in $L^*a^*b^*$ color space

Table 1. Experimental setup for the FBADE algorithm

| Parameter | Value |
|-------------------|----------|
| NP | $10 * D$ |
| k_1, k_2 | [0.5, 1] |
| k_3, k_4 | [0.5, 1] |
| No. of Iterations | 200 |
| No. of Runs | 20 |

Table 2. Average values of performance evaluation metrics computed on 300 images of BSD300 dataset

| Method | PRI | VoI | GCE | BDE |
|--------|---------------|---------------|---------------|---------------|
| FCM | 0.6632 | 3.3827 | 0.4794 | 10.0347 |
| FBADE | 0.6551 | 3.2367 | 0.4472 | 9.9841 |

Table.2. shows average values of the PRI, VoI, GCE and BDE computed on all the 300 images of BSD300 dataset. The best in the class results are marked in bold. FBADE outperforms FCM in three out of the four benchmarks (VoI, GCE and BDE) and even PRI results are pretty close to each others. Hence establishes the supremacy of the proposed algorithm. The distribution of performance measures using FCM and FBADE are displayed in Fig. 2. and Fig. 3. respectively. Few of the segmented images from the BSD300 dataset are displayed in Fig. 4.

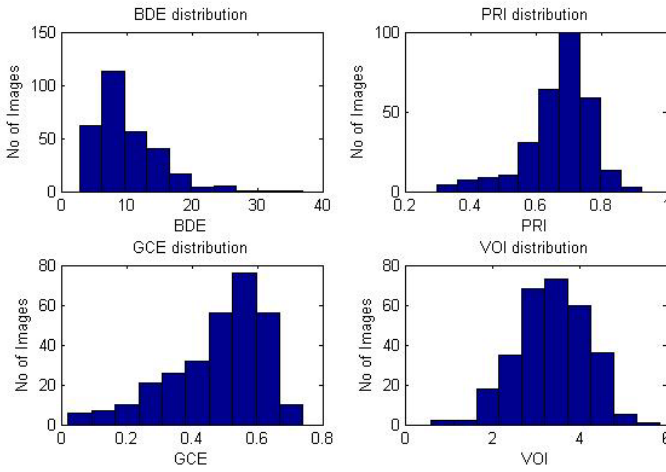


Fig. 2. Distribution of the performance measures over the 300 images of the Berkeley database using FCM

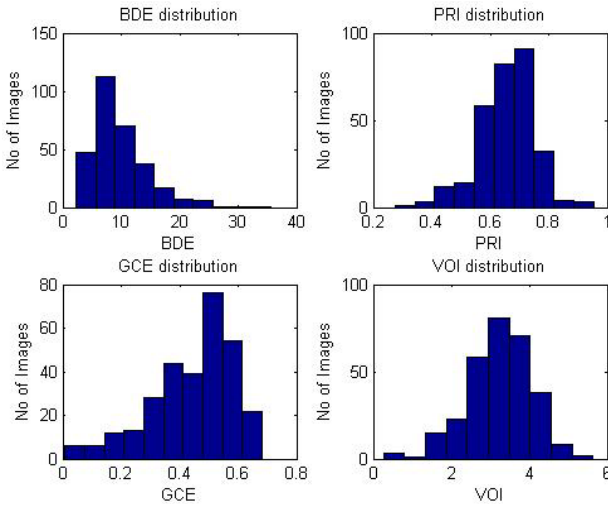


Fig. 3. Distribution of the performance measures over the 300 images of the Berkeley database using FBADE

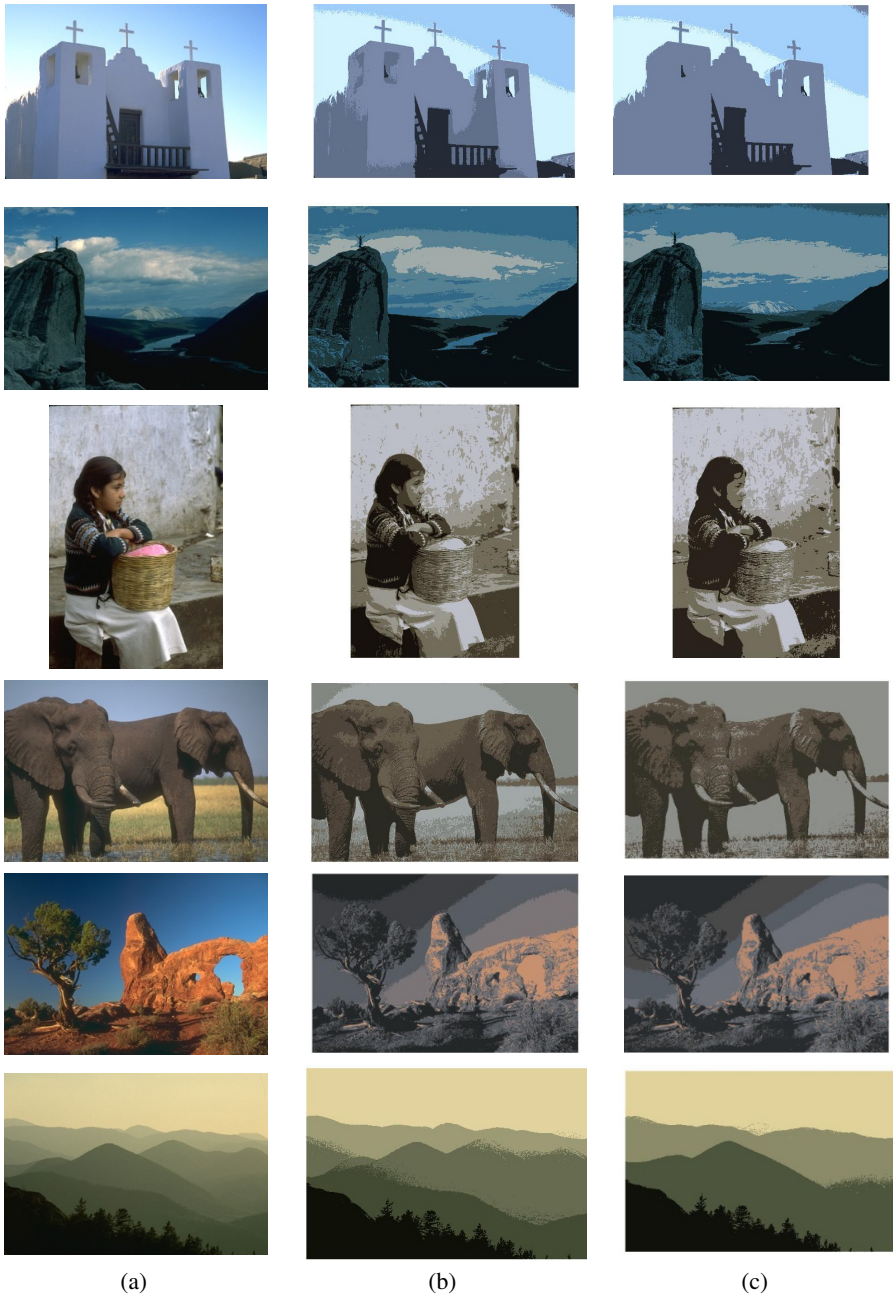


Fig. 2. Results of some example images of BSD 300 dataset (a) original color image, (b) segmented images using FCM and (c) segmented images using FBADE

6 Conclusion

In this paper we have proposed a pixel clustering scheme based on FBADE. Results are computed using well known The Berkley dataset and benchmarks and compared with most popular and widely used clustering technique FCM. In the FBADE approach the adaptation of the control parameters of DE are not predefined, rather they are they are determined adaptively for each solution of the population and the adaptation is based on the fitness statistics of the population. Thus the FBADE prevents the algorithm to get stuck in local minima. The usage of FBADE in this pixel clustering application clearly shows improvements of results when compared with FCM.

In this paper we have considered four performance metrics used for evaluating the quality of the segmentation of BSD300 dataset. However several other validation techniques like Normalized PRI (NPRI), F-measures can be used to test the segmentation results. One could associate a cluster validity index along with fuzzy clustering to form a multi-objective fitness function. In such cases multi-objective DE can be employed to achieve better clustering.

References

- [1] Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. *ACM Computing Surveys* 31(3), 264–323 (1999)
- [2] Tou, J.T., Gonzalez, R.C.: *Pattern Recognition Principles*. Addison-Wesley, London (1974)
- [3] Bezdek, J.C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, New York (1981)
- [4] Ball, G., Hall, D.: A clustering technique for summarizing multivariate data. *Behav. Sci.* 12, 153–155 (1967)
- [5] Scholkopf, B., Smola, A.J.: *Learning with Kernels*. The MIT Press, Cambridge (2002)
- [6] Trivedi, M.M., Bezdek, J.C.: Low-level segmentation of aerial images with fuzzy clustering. *IEEE Trans.on Systems, Man and Cybernetics* 16(4) (July 1986)
- [7] Al-Sultan, K.S., Fedjki, C.A.: A tabu search-based algorithm for the fuzzy clustering problem. *Pattern Recognition* 30, 2023–2030 (1997)
- [8] Sun, L.-X., Danzer, K.: Fuzzy cluster analysis by simulate annealing. *Journal of Chemometrics* 10, 325–342 (1996)
- [9] Lukashin, V., Fuchs, R.: Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters. *Bioinformatics* 17, 405–414 (2001)
- [10] Kuncheva, L.I., Bezdek, J.C.: Selection of cluster prototypes from data by a genetic algorithm. In: *Proc. 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT)*, Aachen, Germany, vol. 18, pp. 1683–1688 (1997)
- [11] Hall, L.O., Ozyurt, I.B., Bezdek, J.C.: Clustering with a genetically optimized approach. *IEEE Transactions on Evolutionary Computation* 3, 103–112 (1999)
- [12] 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)
- [13] 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)

- [14] Das, S., Abraham, A., Konar, A.: Automatic clustering using an improved differential evolution algorithm. *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans* 38(1), 218–237 (2008)
- [15] Maulik, U., Saha, I.: Automatic Fuzzy Clustering Using Modified Differential Evolution for Image Classification. *IEEE Trans. on Geoscience and Remote Sensing* 48(9), 3502–3510 (2010)
- [16] Patra, G.R., Singha, T., Choudhury, S.S., Das, S.: A Fitness-Based Adaptive Differential Evolution Approach to Data Clustering. In: Satapathy, S.C., Udgata, S.K., Biswal, B.N. (eds.) *Proceedings of Int. Conf. on Front. of Intell. Comput. AISC*, vol. 199, pp. 469–475. Springer, Heidelberg (2013)
- [17] Liu, J., Lampinen, J.: On setting the control parameter of the differential evolution method. In: *Proceedings of the 8th International Conference on Soft Computing (MENDEL)*, Brno, Czech Republic, pp. 11–18 (2002)
- [18] Zaharie, D.: Critical values for the control parameters of differential evolution algorithms. In: *Proceedings of MENDEL 2002, 8th International Mendel Conference on Soft Computing*, Bruno, pp. 62–67 (2002)
- [19] Price, K., Storn, R., Lampinen, J.: *Differential Evolution: A Practical Approach to Global Optimization*. Springer (2005)
- [20] Liu, J., Lampinen, J.: A fuzzy adaptive differential evolution algorithm. *Soft Computing* 9(6), 448–462 (2005)
- [21] Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pp. 1785–1791 (2005)
- [22] Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation* 13, 398–417 (2009)
- [23] Brest, J., Greiner, S., Bošković, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark functions. *IEEE Transactions on Evolutionary Computation* 10(6), 646–657 (2006)
- [24] 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)
- [25] Islam, S.M., Das, S., Ghosh, S., Roy, S., Suganthan, P.N.: An Adaptive Differential Evolution Algorithm With Novel Mutation and Crossover Strategies for Global Numerical Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 42(2), 482–500 (2012)
- [26] 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)
- [27] Srinivas, M., Patnaik, L.M.: Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 24(4), 656–667 (1994)
- [28] Yang, A.Y., Wright, J., Ma, Y., Sastry, S.S.: Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding* 110, 212–225 (2008)

Performance Study of a New Modified Differential Evolution Technique Applied for Optimal Placement and Sizing of Distributed Generation

S. Kumar, D. Pal, K. K. Mandal, and Niladri Chakraborty

Power Engineering Department, Jadavpur University, (2nd Campus), Salt Lake III,
Kolkata-700098, India

{sajjan.pradhan48, dptndpal0}@gmail.com,
kkm567@yahoo.co.in, chakraborty_niladri2004@yahoo.com

Abstract. Determination of optimal placement and sizing of Distributed Generations (DGs) is one of the important tasks in power system operation. Several conventional as well as heuristics techniques like particle swarm optimization, differential evolution etc. have been applied to solve the problem. But one of the major drawback of these techniques are the improper selection of user defined parameters for optimal solution. Improper selection of the parameters may even lead to premature convergence. A new modified differential evolution technique based algorithm is proposed in this paper for the solution of optimal sizing and location of distributed generation to avoid premature convergence. The proposed algorithm is applied on IEEE 14 and 30 bus systems to verify its effectiveness. The results obtained by the proposed method are compared with other methods. It is found that the results obtained by the proposed algorithm are superior in terms of cost and losses.

Keywords: Distributed Generation, Sizing and placement, Voltage Profile, Modified Differential Evolution.

1 Introduction

Structure of electrical power system is undergoing a profound change because of several limitations of conventional centralized power plants viz. high transmission losses, huge environmental impacts, fuel unavailability etc. New trend of electrical power system incorporates Distributed Generation (DG). Different organizations/researchers defines the DG in different ways like Electric Power Research Institute [EPRI] defines distributed generation as generation from a few kilowatts up to 50 MW [1], The International Conference on Large High Voltage Electric Systems (CIGRE) defines DG as smaller than 50–100 MW [2], International Energy Agency (IEA) defines distributed generation as generating plant serving a customer on-site or providing support to a distribution network, connected to the grid at distribution-level voltages [3], Ackermann et al [4] suggested different ratings after consideration the technical issues related with DG like micro DG: 1 to 5 KW, small

DG: 5 KW to 5 MW, medium DG: 5 MW to 50 MW and large DG: 50 to 300 MW. In other words, it can be defined as “DG has many small-scale generators which are installed on various strategic points throughout the system depending upon the local load demand to minimize the electricity losses and maximize the reliability”. The technology adopted in DG may be renewable or/and nonrenewable sources of energy like wind generators, photo voltaic (PV) cell, mini/micro hydro power plant, gas turbines, fuel cell, combined cycle plants etc. depending on the system structure, resources availability and system reliability. DG can be used as the stand alone system or grid integrated system.

Optimal sizing and placement of DG is one of the most essential conditions for minimizing the losses and cost, maximizing the reliability and improving the voltage profile. Several researchers have worked in this area and they have proposed different techniques for the same.

A conventional iterative search technique with Newton-Raphson method has been used to find the optimal sizing and placement of DG [5]. The authors found the Weighting Factor (WF) first and then this WF was used to calculate the DG location and its sizing in two different steps. Optimal sizing and placement of different types of DG by conventional power flow method was presented by Abri *et al.* [6]. The main goal was to utilize the DG units to improve the voltage stability margin considering the probabilistic nature of both loads and renewable DG generation. Improved particle swarm optimization (IPSO) algorithm and Monte Carlo simulation based multi-objective optimization technique was also applied for determination of optimal location and size of DGs [7]. Prenc *et al.* discussed the optimal allocation of three types of distributed generation units with a goal of minimizing cumulative average daily active power losses by the use of Genetic Algorithm (GA) [8]. Solar parks, wind farms and power stations which are not depend on an intermittent primary energy source, are taken into consideration. A modified Teaching-Learning Based Optimization (TLBO) algorithm has been used [9] to find the best sites and size to connect DG systems in a distribution network. Nayak *et al.* applied Differential Evolution Algorithm (DEA) to find the optimal placement and sizing of DG in the IEEE-69 bus radial distribution system [10]. Here the main objective was to minimize the total real power loss and improve the voltage profile within the frame work of system operation and security constraints. A multi-objective Real Coded Genetic Algorithm is also used [11] to find the optimal location and sizing for minimization of DG cost and line loss. The work has been done by two methods. The first method gave only one compromised solution considering both the objectives whereas the second method provided a very close suboptimal solution by considering both the objectives simultaneously by creating set of pareto optimal solutions for the congestion problem.

In an attempt to improve the optimum results, this paper presents the application of a new Modified Differential Evolution (MDE) optimization technique in the field of distributed generation, to find the optimal size and location of generator unit. This technique has better control strategy due to random localization of mutation factor and selecting the best candidate as compared with random selection in basic Differential Evolution (DE). A comparative analysis on the performance of different

values of crossover constant (CR) is also studied. This technique is applied on IEEE-14 bus and IEEE-30 bus test systems for performance study.

2 Problem Formulation

The main objective of optimal sizing and placement of distributed generations is to minimize the overall system cost as well as loss by the installation of DG unit in the network system. For the calculation of optimal DG size, first of all, a load flow analysis program is used for calculation of all unknown parameters like voltage magnitude $|V|$ and voltage angle δ for PQ Bus or Load Bus; and voltage angle δ and reactive power Q for PV Bus or Generator Bus. After calculating the unknown parameters, the *Line Current Flow* I_{ij} can be easily calculated by using (1).

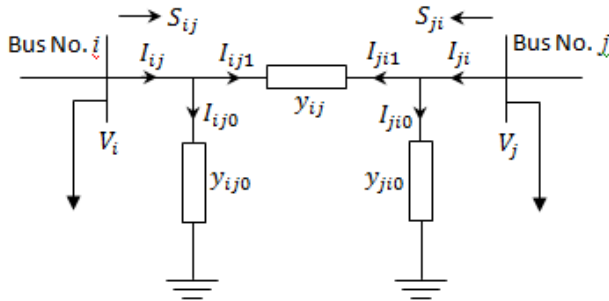


Fig. 1. Network model

$$I_{ij} = I_{ij1} + I_{ij0} = (V_i - V_j)y_{ij} + (V_i \cdot y_{ij0}) . \tag{1}$$

Where,

$i, j = 1, 2, \dots$, No. of Bus,

I_{ij} is a phasor current which flows from bus i to j ,

I_{ij1} & I_{ij0} are the series and shunt current respectively.

Voltage of i^{th} bus, $V_i = |V_i|(\cos\delta_i + j \sin\delta_i)$

And, y_{ij} & y_{ij0} are the series and shunt admittance between line i and j .

Line Power Flow from i^{th} to j^{th} bus (S_{ij}) can be calculated as

$$S_{ij} = V_i \cdot I_{ij}^* = V_i \{ [V_i^* - V_j^*] y_{ij}^* + V_i^* y_{ij0}^* \} . \tag{2}$$

Similarly,

$$S_{ji} = V_j \{ [V_j^* - V_i^*] y_{ji}^* + V_j^* y_{ji0}^* \} . \tag{3}$$

After that, *Total Line Losses* (TLL) can be calculated using (4).

$$TLL = \sum_{i=1}^{bus\ no.} \sum_{j=1}^{bus\ no.} (S_{ij} + S_{ji}). \quad (4)$$

The main objective of this work is to minimize the overall cost of the system by means of minimizing the total line loss (TLL). So, the Objective Function (OF), which is to be minimized, can be expressed as

$$OF = \min (C_{DG} + W * real (TLL)). \quad (5)$$

$$OF = \min \left[C_{DG} + W * real \left(\sum_{i=1}^{bus\ no.} \sum_{j=1}^{bus\ no.} (S_{ij} + S_{ji}) \right) \right]. \quad (6)$$

Where,

$$C_{DG} = a_{DG} + b_{DG} \cdot P_{DG} + c_{DG} \cdot (P_{DG})^2. \quad (7)$$

C_{DG} is total cost of DG as a function of DG rating (P_{DG}), W is weighting factor, a_{DG} , b_{DG} and c_{DG} are the quadratic cost coefficients of DG. Weighting factor (W) is a constant for the system, which converts the loss into their respective cost.

Various soft-computing optimization techniques can be applied for minimization of objective function. But among those techniques, modified differential evolution (MDE) optimization technique is used in this paper because it is one of the robust optimization technique in which very few user dependent parameter is used. It is seen that MDE has very good convergence characteristic and also it doesn't stagnate to local minima.

3 Modified Differential Evolution Technique

The Modified Differential Evolution (MDE) technique is an advanced version of classical Differential Evolution (DE) technique. The classical Differential Evolution technique was first introduced by Storn and Price in the year 1995 [12], [13]. In MDE, the mutation process has been modified for improving the convergence characteristics and decreasing the computational time. This MDE algorithm had been first proposed by Kaelo and Ali [14]. It is also a population based iterative optimization technique which is stochastic in nature. Similar to Genetic Algorithm (GA), MDE algorithm also uses crossover, mutation and selection operators. But, Simple GA uses a binary coding for representing problem parameters whereas MDE uses real coding of floating point numbers. In MDE, mutation process is applied to generate a trial vector, which is then used within the crossover process to produce one offspring; hence all the solutions have the same chance of being selected as parents. In DE algorithm, there are mainly three control parameters, which are differentiation (mutation) constant F , crossover constant CR , and size of population NP ; whereas, MDE technique don't use mutation constant F as a control parameter. In MDE, this mutation parameter is not user controlled. It is chosen randomly within the range $[-1, -0.4] \cup [0.4, 1]$. Maximum number of generations or iterations (G) is taken as

stopping criteria. The dimension of the problem (D) depends on the problem's unknown variables, which directly affects the difficulty of optimization task and computational time to converge the solution. The MDE algorithm works through a simple cycle of stages, which are initialization, mutation, crossover and selection. Their flow patterns are given below:

```

Initialization
Evaluation
Repeat
  Mutation
  Crossover
  Evaluation
  Selection
  Increase the generation count
Until (Generation count doesn't exceed the maximum
limit)

```

3.1 Initialization

At the beginning of the solution methodology, first of all, problem independent variables are initialized according to their maximum and minimum value. These newly generated vectors are called target vectors or parent vectors ($x_{i,j}^k$) and this can be calculated as,

$$x_{i,j}^k = x_{min,j} + \sigma_{i,j} \cdot (x_{max,j} - x_{min,j}) . \quad (8)$$

Where,

$i=1, 2, \dots, NP, j = 1, 2, \dots, D$.

k indicates the generation or iteration number. For initialization process, $k = 1$.

$x_{min,j}$ and $x_{max,j}$ are the minimum and maximum limit of j^{th} variable.

$\sigma_{i,j}$ is the random numbers such that $\sigma_{i,j} \in [0, 1]$.

3.2 Mutation

In each generation, this process creates donor vectors ($v_{i,j}^k$) by mutating the target vectors. To create a donor vector, three random numbers r_1, r_2 and r_3 are taken such that $r_1 \neq r_2 \neq r_3 \neq i$. With the help of these numbers, in each generation, three random parameter vectors $x_{r_1,j}, x_{r_2,j}$ and $x_{r_3,j}$ are chosen for all dimension (i.e. $j = 1, 2, \dots, D$), from the current population or target vectors.

In the classical DE, the mutation process is done between three random vectors and the base vector is then chosen at random within the three. The new donor vector for the j^{th} component of each vector is expressed as,

$$v_{i,j}^k = x_{r_1,j}^k + F \cdot (x_{r_2,j}^k - x_{r_3,j}^k) . \quad (9)$$

Theoretically, the mutation constant F can be anything between 0 to ∞ [15]; but, usually it is taken between the range of 0.4 to 1.0 [16]. The value of k depends on iteration number which varies from 1 to maximum number of generation (G).

Due to this random localization of vectors, it has an exploratory effect but it slows down the convergence of DE. So, the first modification to DE is to replace the random base vector (x_{r1}^k) in (9) with the tournament best (x_{tb}^k). From the three random vectors, the best is used as the base vector and the remaining two are used to find the differential vector in (9). So, the region around each x_{tb}^k explores for each mutated point. This maintains the exploratory feature and at the same time expedites the convergence. Also the original DE uses the fixed mutation factor F in mutation process but in this modified version of DE, it uses a random F in $[-1, -0.4] \cup [0.4, 1]$ for each mutated point. This also improves the exploration. This version of DE is referred to as the differential evolution algorithm with random localization (DERL) [14]. Thus the modified mutation process can be expressed as:

$$v_i^k = x_{tb}^k + F_i \cdot (x_{r2}^k - x_{r3}^k) . \quad (10)$$

Where, $F_i \in [-1, -0.4] \cup [0.4, 1]$, chosen randomly.

3.3 Crossover

This operation is done for increasing the diversity of the population. In this process, the donor vector exchanges its components with those of the current (parent) member $x_{i,j}^k$ and forms trial or child vectors $u_{i,j}^k$ that will compete with the parent vector $x_{i,j}^k$. The new trial vector can be expressed as:

$$u_{i,j}^k = \begin{cases} v_{i,j}^k & \text{if } (\mu_i \leq CR) \\ x_{i,j}^k & \text{else.} \end{cases} \quad (11)$$

Where, μ_i denotes a uniformly distributed random number within the range $[0, 1]$, generated a new for each value of i . The crossover constant CR usually taken from within the range $[0, 1]$, but for better control of diversity and prevention for stagnation of solution to local minima, it varies within the range $[0.8, 0.9]$.

3.4 Selection

This process is carried out to determine which one of the trial vector and the target vector will survive in the next generation, i.e. at generation $k = k + 1$. Survival of the vector (trial or target) depends on the fittest vector for the objective function, i.e. for the next generation, target vector will be replaced by trial vector if and only if the trial vector yields a better value of objective function or fitness value. So, for the generation of the new target vector, selection process can be expressed as,

$$x_i^{k+1} = \begin{cases} u_i^k & \text{if } f(u_i^k) \leq f(x_i^k) \\ x_i^k & \text{else.} \end{cases} \quad (12)$$

Where, $f(\cdot)$ indicates the objective function (OF), which is to be minimized.

4 Results and Discussion

The proposed algorithm is applied on IEEE-14 and 30 bus test systems to verify its effectiveness and implemented with MATLAB on a PC of Dual Core, 1.73 GHz Processor, 3 GB of RAM. The system data for IEEE-14 bus and 30 bus are taken from [17], [18] and [18], [19] respectively. The cost coefficients of DG are taken from [20]. For the present MDE optimization technique, population size is taken as 30 and the maximum number of iteration is taken as 100. The optimal value for the crossover factor obtained as 0.9 and 0.8 for the IEEE-14 and 30 bus systems respectively. These values were selected through trial and error methods. The best results obtained by this technique are shown in Table 1. It is found from the Table 1 that the optimal DG size is 34.12 MW and located at bus number 3 for IEEE 14 bus system. Optimal active power loss and cost are found to be 11.54 MW and \$ 1989.4 for the same system respectively. The optimal DG size for the IEEE-30 bus system is found to be 49.96 MW and located at bus number 5, whereas the loss and cost are found to be 13.32 MW and \$ 2561.0 respectively. The computation time are found to be 29.32 sec. and 129.14 sec. for the IEEE-14 and 30 bus respectively.

Table 1 also compared the results with Conventional Iterative Search technique with N-R Load Flow Method and Power World Simulator [5]. It is found that the proposed technique can produce superior results in terms of both loss and cost. After applying MDE optimization technique, for IEEE 14 bus system, 1.536 % and 1.368 % total active power loss is reduced, whereas, for IEEE 30 bus system, the reduction in total active power loss is increased to 3.198 % and 2.131 % as compared to Power world Simulator and Conventional Iterative Search technique with N-R Load Flow Method respectively. In the term of total system cost, the MDE optimization technique produces about 0.43 % and 0.28 % superior results for IEEE 14 bus system whereas about 1.08 % and 0.21 % superior results for IEEE 30 bus system as compared to Power world Simulator and Conventional Iterative Search technique with N-R Load Flow Method respectively.

Table 1. Results obtained by the proposed method and its comparative study

| IEEE Test Systems | IEEE-14 Bus | IEEE-30 Bus |
|---|---------------|---------------|
| DG Size (MW) | 34.12 | 49.96 |
| Placement of DG (Bus No.) | 3 | 5 |
| Power World Simulator[5] | 11.72 | 13.76 |
| Total Active Loss (MW) Conventional Iterative Search technique with N-R Load Flow Method [5] | 11.70 | 13.61 |
| MDE | 11.54 | 13.32 |
| Power World Simulator[5] | 1998.00 | 2589.00 |
| Optimal Cost (\$) Conventional Iterative Search technique with N-R Load Flow Method[5] | 1995.00 | 2566.50 |
| MDE | 1989.4 | 2561.0 |

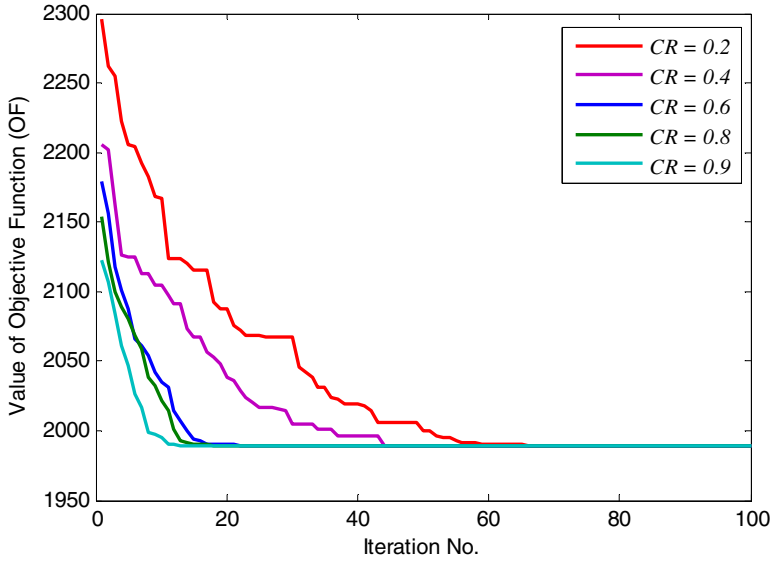


Fig. 2. Variation of optimal cost with iteration no. for IEEE 14 bus system

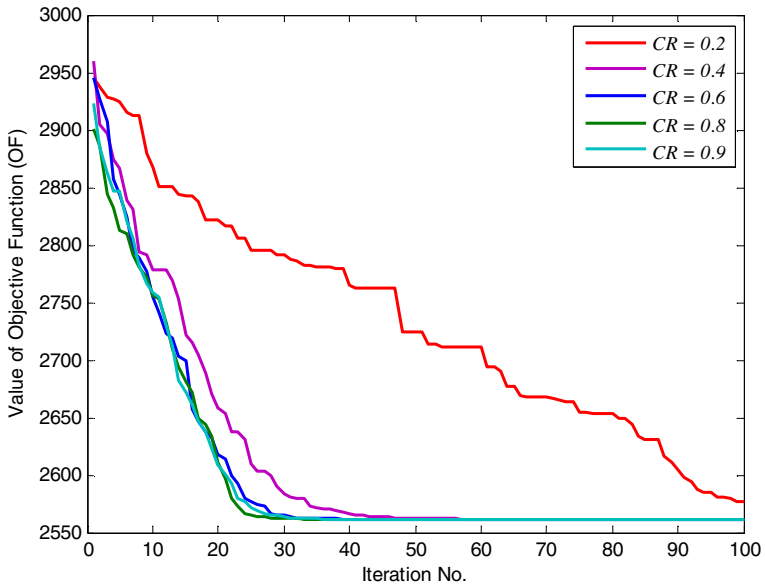


Fig. 3. Variation of optimal cost with iteration no. for IEEE 30 bus system

Fig. 2 and Fig. 3 show the convergence characteristic for optimal cost for IEEE-14 bus and 30 bus respectively. The effects of different values of crossover ratios (CR) are also studied for the two systems and the results are shown in Fig. 2 and Fig. 3. From both the figures, it can be easily noticed that convergence characteristics for optimal cost are very unsatisfactory for lower value of CR and prematurely converges to suboptimal values. The values of CR the range 0.8 to 0.9 give very good results.

5 Conclusion

In this work, a new modified differential evolution based algorithm is proposed for the optimal sizing and location of distributed generation systems. The proposed algorithm is applied on IEEE14 and 30 bus systems. It is found that proposed algorithm can avoid premature convergence and can produce optimal results. The results obtained by the proposed techniques are compared with other methods and found to produce superior results both in terms of cost and losses.

Acknowledgments. This work is supported by University Grants Commission (UGC) under the scheme of University with Potential for Excellence, Phase – II (UPE – II) and Departmental Research Scheme (DRS) granted to Power Engineering Department, Jadavpur University.

References

1. Electric Power Research Institute, <http://www.epri.com/gg/newgen/disgen/index.html>
2. CIGRE, “Impact of increasing contribution of dispersed generation on the power system”, CIGRE Study Committee no 37, Final Report (September 1998)
3. Distributed Generation in Liberalized Electricity Markets, International Energy Agency/OECD. IEA, Paris, p. 19 (2002)
4. Ackermann, T., Andersson, G., Sodder, L.: Distributed generation: a definition. *Elect. Power Syst. Res.* 57, 195–204 (2001)
5. Ghosh, S., Ghoshal, S.P., Ghosh, S.: Optimal sizing and placement of distributed generation in a network system. *Electrical Power and Energy Systems* 32, 849–856 (2010)
6. Al Abri, R.S., El-Saadany, E.F., Atwa, Y.M.: Optimal Placement and Sizing Method to Improve the Voltage Stability Margin in a Distribution System Using Distributed Generation. *IEEE Trans. on Power Systems* 28(1) (February 2013)
7. Abdi, S., Afshar, K.: Application of IPSO-Monte Carlo for optimal distributed generation allocation and sizing. *Electrical Power and Energy Systems* 44, 786–797 (2013)
8. Prenc, R., Škrlec, D., Komen, V.: Distributed generation allocation based on average daily load and power production curves. *Electrical Power and Energy Systems* 53, 612–622 (2013)
9. García, J.A.M., Mena, A.J.G.: Optimal distributed generation location and size using a modified teaching–learning based optimization algorithm. *Electrical Power and Energy Systems* 50, 65–75 (2013)

10. Nayak, M.R., Dash, S.K., Rout, P.K.: Optimal Placement and Sizing of Distributed Generation in Radial Distribution System Using Differential Evolution Algorithm. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Nanda, P.K. (eds.) SEMCCO 2012. LNCS, vol. 7677, pp. 133–142. Springer, Heidelberg (2012)
11. Vijayakumar, K., Jegatheesan, R.: Optimal Location and Sizing of DG for Congestion Management in Deregulated Power Systems. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Nanda, P.K. (eds.) SEMCCO 2012. LNCS, vol. 7677, pp. 679–686. Springer, Heidelberg (2012)
12. Storn, R., Price, K.: Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. Berkeley (CA): International Computer Science Institute. Technical report TR-95-012 (1995)
13. Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization* 11, 341–359 (1997)
14. Kaelo, P., Ali, M.M.: A numerical study of some modified differential evolution algorithms. *European Journal of Operational Research* 169(3), 1176–1184 (2006)
15. Amjady, N., Sharifzadeh, H.: Solution of non-convex economic dispatch problem considering valve loading effect by a new Modified Differential Evolution algorithm. *Electrical Power and Energy Systems* 32, 893–903 (2010)
16. El Ela, A.A.A., Abido, M.A., Spea, S.R.: Differential evolution algorithm for optimal reactive power dispatch. *Electric Power Systems Research* 81, 458–464 (2011)
17. Pai, M.A.: *Techniques in power system analysis*. Tata McGraw-Hill Publishing Company Limited, New Delhi (2006)
18. Wallach, Y.: *Calculations & Programs for Power System Networks*. Prentice-Hall, Inc., Englewood Cliffs
19. Yokoyama, R., Bae, S.H., Morita, T., Sasaki, H.: Multiobjective optimal generation dispatch based on probability security criteria. *IEEE Trans. on Power Systems* 3, 317–324 (1988)
20. Durga, G., Nadarajah, M.: Optimal DG placement in deregulated electricity market. *Electric. Power Syst. Res.* 77, 1627–1636 (2007)

An Approach to Solve Multi-criteria Supplier Selection While Considering Environmental Aspects Using Differential Evolution

Sunil Kumar Jauhar¹, Millie Pant¹, and Aakash Deep²

¹ Indian Institute of Technology, Roorkee, India

² Jaypee University of Engineering and Technology, Guna, India
{suniljauhar.iitr,millidma,aakash791}@gmail.com

Abstract. Selection of an appropriate supplier is gaining a lot interest among researchers working in the field of supply chain management. Often many suppliers are available in the market that fulfill some preliminary criteria. However the real task is to determine the most suitable set of suppliers (or key suppliers) subject to management as well as environmental aspects. In the current study, we present an approach to solve the multiple-criteria green supplier selection problem (mathematical model formulated with Data Envelopment Analysis) with the application of differential evolution. A hypothetical case demonstrates the application of the present approach.

Keywords: Green supplier selection, Supply chain management, Differential evolution, Data Envelopment Analysis, Multi criteria decision making, CO₂ Emissions.

1 Introduction

Supplier selection is an important part of supply chain management (SCM). Traditionally, only the management aspects like lead time, quality and price of the supply chain were considered for selecting a potential supplier. However, with the growing environmental issues researchers are also paying attention to factors like greenhouse effect, carbon-di-oxide (CO₂) emission etc. jointly known as Carbon foot printing. The resulting problem is called '*green supplier selection*', where a balance is maintained between the management and environment issues.

The supplier selection practices are extensively studied in the literature with multi-criteria decision analysis models (MCDM). These models comprise approaches, as Data envelopment analysis (DEA), analytic hierarchy process (AHP), or analytic network process (ANP) etc [1]. Investigations focus on the environmental aspects of the supplier selection and evaluation among them a number of papers attempt to build in green criteria into the criteria of supplier selection [2,3] only few examinations (e.g. [4]) relate to such situations, when it is not possible to practice sophisticated methodology e.g. because the lack of strong mathematical background.

Differential evolution (DE) algorithm proposed by Storn and Price in 1995 [5] is a population set based evolutionary algorithm that has been applied successfully to a wide range of problems [6-10]. In the present study, DE is used for solving the green supplier selection problem modeled with the help of DEA. The aim of this paper is to contribute to green supplier selection approach. In present approach we introduce the green criteria such as reusability and carbon emission in the supplier selection.

This paper is organized in seven sections. Subsequent to the introduction in Section 1, the green supplier selection and Problem statement and methodology are briefed in Sections 2 and 3. Section 4 describes the Mathematical Formulation of the Problem with DEA used in this paper. Section 5 describes the DE algorithm for green supplier selection. Finally, a discussion and Conclusion drawn from the present study are given in Section 6 and 7.

2 Green Supplier Selections

In recent years, an increasing environmental awareness has favored the emergence of the new Green Supply Chain paradigm; thus, also in the Supplier selection problem, green criteria have been incorporated. The Green Supplier Selection Problem can be defined as “a classical Supplier selection problem in which, among the others, also environmental criteria are taken into account in order to select and monitor suppliers’ performances” [11].

In current study we split the environmental criteria as reusability and CO₂ emissions of the product and services. We assume that the environmental criteria are the outputs of the examined model. In the current study, a supplier is considered efficient if the efficiency score is 1 otherwise it is considered as inefficient.

3 Problem Statements and Methodology

This hypothetical case of an Indian automobile part manufacturing company at northern part of India is presented here to illustrate the present approach. In the present study we have considered a case of selecting best suppliers out of 18 potential suppliers. The criteria considered for selection as given below:

1. Management criteria: Lead time, quality and price
2. Environmental (green) criteria: Reusability and CO₂ emission

The hypothetical data range for Management criteria (Inputs) as well for Environmental criteria (Output) are given below;

Management criteria,

- | | |
|--------------------|--------------|
| 1. Lead time (Day) | =1-5 days |
| 2. Quality (%) | = 50%-100% |
| 3. Price (Rs.) | =100-300 Rs. |

Environmental criteria,

1. Reusability (%) ($L_R = L_M - L_A$) =30%-70%
2. CO₂ Emissions (g) =10-30g

Reusability concept is taken from [12] and for CO₂ Emissions, *LOCOG* Guidelines on Carbon Emissions of Products and Services –Version 1 [13] is considered.

The underlying data is shown in Table 1 with the supplier's database covering management as well as environmental criteria of an item provided in the shipment of automobile company.

Table 1. Supplier's database

| Criteria | Management criteria (Inputs) | | | Environmental criteria (Outputs) | |
|-----------|------------------------------|-----------------|-----------------|----------------------------------|------------------------------------|
| | Lead time (L) (Day) | Quality (Q) (%) | Price (P) (Rs.) | Reusability(R) (%) | CO ₂ Emissions (CE) (g) |
| Suppliers | | | | | |
| 1 | 2 | 80 | 107 | 70 | 30 |
| 2 | 1 | 70 | 161 | 50 | 10 |
| 3 | 3 | 90 | 269 | 60 | 15 |
| 4 | 4 | 65 | 270 | 30 | 12 |
| 5 | 2 | 55 | 260 | 40 | 18 |
| 6 | 5 | 70 | 201 | 50 | 20 |
| 7 | 3 | 85 | 111 | 66 | 14 |
| 8 | 2 | 95 | 300 | 35 | 28 |
| 9 | 1 | 67 | 197 | 60 | 16 |
| 10 | 4 | 72 | 157 | 44 | 28 |
| 11 | 5 | 51 | 170 | 41 | 14 |
| 12 | 3 | 58 | 106 | 49 | 10 |
| 13 | 2 | 72 | 255 | 32 | 25 |
| 14 | 4 | 60 | 117 | 40 | 29 |
| 15 | 5 | 63 | 245 | 22 | 5 |
| 16 | 3 | 90 | 299 | 10 | 9 |
| 17 | 1 | 87 | 101 | 42 | 15 |
| 18 | 2 | 82 | 206 | 70 | 18 |

3.1 Methodology

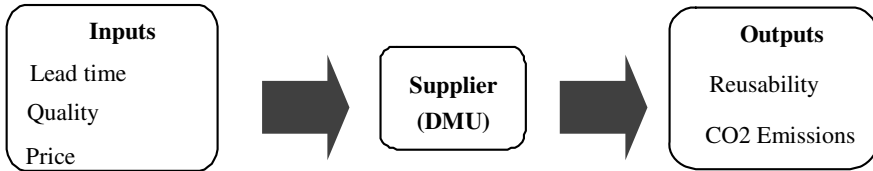
To measure and analyze the relative efficiency of 18 suppliers, we follow a four step methodology:

- Design a criteria containing environmental and management aspects.
- Select a problem
- Formulate the mathematical model of the problem with the help of DEA.
- Apply DE on mathematical model.

By using this methodology, the company can obtain a recommended combination of efficient suppliers.

4 Mathematical Formulation of the Problem with DEA

DEA based method is used for determining the efficiency of Decision-Making Unit (DMU) on the basis of multiple inputs and outputs [14]. DMU can comprise of manufacturing units, departments of big organizations such as universities, schools, hospitals, power plants, police stations, tax offices, prisons, a set of firms etc [15]. The DMU well-defined in this study with input and output criteria are as follows:



The performance of DMU is estimated in DEA by the concept of efficiency or productivity, which the ratio of weights sum of outputs to the weights sum of inputs [16] i.e

$$Efficiency = \frac{\text{Weighted sum of outputs}}{\text{Weighted sum of inputs}} \tag{1}$$

The two basic DEA models are the CCR (Charnes, Cooper and Rhodes) model [17] and the BCC (Banker, Charnes and Cooper) model [18], these two models distinguish on the returns to scale assumed. The former assumes constant returns-to-scale whereas the latter assumes variable returns-to-scale [14]. In the current study we use CCR model which is well-defined further down:

Assume that there are N DMUs and each unit have I input and O outputs then the efficiency of m^{th} unit is obtained by solving the following model which is proposed by Charnes et al [17].

$$\text{Max } E_m = \frac{\sum_{k=1}^O w_k \text{Output}_{k,m}}{\sum_{l=1}^I z_l \text{Input}_{l,m}} \tag{2}$$

$$0 \leq \frac{\sum_{k=1}^O w_k \text{Output}_{k,n}}{\sum_{l=1}^I z_l \text{Input}_{l,n}} \leq 1; n = 1, 2, \dots, m \dots N$$

Where

- E_m is the efficiency of the m^{th} DMU, $k=1$ to O , $l=1$ to I and $n=1$ to N .
- $Output_{k,m}$ is the k^{th} output of the m^{th} DMU and w_k is weight of output $Output_{k,m}$
- $Input_{l,m}$ is the l^{th} input of m^{th} DMU and z_l is the weight of $Input_{l,m}$
- $Output_{k,n}$ and $Input_{l,n}$ are the k^{th} output and l^{th} input respectively of the n^{th} DMU

The fractional program shown in Equ-2 can be converted in a linear program which is shown in Equ-3

$$\begin{aligned}
 & \text{Max} \quad E_m \sum_{k=1}^O w_k Output_{k,m} \\
 & \text{s.t.} \\
 & \sum_{l=1}^I z_l Input_{l,m} = 1 \\
 & \sum_{k=1}^O w_k Output_{k,n} - \sum_{l=1}^I z_l Input_{l,n} \leq 0, \quad \forall n \\
 & w_k, z_l \geq 0; \quad \forall k, l
 \end{aligned} \tag{3}$$

To determine the efficiency score of each DMU we run the above program run N times. A DMU is considered efficient if the efficiency score is 1 otherwise it is considered as inefficient.

4.1 Mathematical model

On the basis of the hypothetical data given in Table 1 the DEA model of m^{th} DMU will be as follows:

$$\begin{aligned}
 & \text{Max} \quad R_m + CE_m \\
 & \text{s.t.} \\
 & z_1 L_m + z_2 Q_m + z_3 P_m = 1 \\
 & w_1 R_n + w_2 CE_n - (z_1 L_m + z_2 Q_m + z_3 P_m) \leq 0 \\
 & \forall n = 1, \dots, m, \dots, 18
 \end{aligned} \tag{4}$$

5 DE Algorithm for Green Supplier Selection

Introduced by Storn and Price in 1995, It is Population-based search technique for global optimization. DE algorithm is a kind of evolutionary algorithm, used to optimize the functions. In current study we have used DE/rand/1/bin scheme [19] and DE algorithm from [20].

5.1 Pseudo Code of SDE Algorithm

```

1   Begin
2   Generate uniformly distribution random population  $P = \{X_{1,G}, X_{2,G}, \dots, X_{NP,G}\}$ .
    $X_{i,G} = X_{i,G}^{lower} + (X_{i,G}^{upper} - X_{i,G}^{lower}) * rand(0,1)$ , where  $i = 1, 2, \dots, NP$ 
3   Evaluate  $f(X_{i,G})$ 
4   While (Termination criteria is met )
5   {
6   For  $i = 1 : NP$ 
7   {
8   Select three random vector  $X_{r1,G}, X_{r2,G}, X_{r3,G}$  where  $i \neq r_1 \neq r_2 \neq r_3$ 
9   Perform mutation operation
10  Perform crossover operation
11  Evaluate  $f(U_{i,G})$ 
12  Select fittest vector from  $X_{i,G}$  and  $U_{i,G}$  to the population of next generation
13  }
14  Generate new population  $Q = \{X_{1,G+1}, X_{2,G+1}, \dots, X_{NP,G+1}\}$ 
15  } /* end while loop*/
16  END

```

5.2 Constraint Handling

For the constraint problems various methods have been suggested in literature. A survey of different methods for constraint handling can be found in [21] and [22]. In this paper Pareto-Ranking method is used for handling the constraints [23].

5.3 Parameter Setting for Differential Evolution Algorithm

In this paper we have applied DE to solve the DEA model. The parameter settings for DE are given in Table-2.

Table 2. Parameter setting for DE

| | |
|-------------------------|------|
| Pop size (NP) | 100 |
| Scale Factor (F) | 0.5 |
| Crossover rate (Cr) | 0.9 |
| Max iteration | 3000 |

The program is implemented in DEV C++ and all the uniform random number is generated using the inbuilt function $rand()$ in DEV C++. The fitness value is taken as the average fitness value in 30 runs and the program is terminate when reach to Max-Iteration.

A purchaser (decision maker) can influence a decision (supplier selection) with the choice of weight system. For this purpose with the help of program which is implemented is DEV C++, we intended to generate all the uniform random number (in between 0 to 1) using the inbuilt function *rand ()* in DEV C++, to assist the selection of the weights for management as well as environment aspects in a way to allow the control the result of the selection process.

6 Results and Discussions

The research of efficient multi-criteria green supplier selection problem can obtain a recommended combination of efficient suppliers 1, 9 and 14 using DE algorithm gives the better solution.

Table 3. Average efficiency and weighted of 18 suppliers in 30 runs

| Suppliers | Value of input and output weight | | | | | Efficiency |
|-----------|----------------------------------|--------------|--------------|----------------|----------------|------------|
| | Z1 | Z2 | Z3 | W ₁ | W ₂ | |
| 1 | 0.100129 | 0.00204427 | 0.00594674 | 0.0124772 | 0.00422648 | 1 |
| 2 | 1 | 0 | 0 | 0.0166683 | 1.08283e-017 | 0.833417 |
| 3 | 0 | 0.0111122 | 7.18753e-016 | 0.0124103 | 4.33556e-01 | 0.744619 |
| 4 | 0 | 0.0153862 | 0 | 0.0165195 | 0.00248747 | 0.525436 |
| 5 | 0 | 0.0181836 | 0 | 0.011392 | 0.0219116 | 0.850091 |
| 6 | 0 | 0.0142871 | 9.19188e-020 | 0.0153397 | 0.00230969 | 0.81318 |
| 7 | 0 | 4.71429e-018 | 0.00900991 | 0.0137737 | 2.58488e-01 | 0.909066 |
| 8 | 0.442549 | 0.00121055 | 0 | 0 | 0.0327347 | 0.916572 |
| 9 | 1 | 0 | 0 | 0.00337086 | 0.0498655 | 1 |
| 10 | 0.0794687 | 0.00947534 | 3.71282e-009 | 1.42912e-008 | 0.0305688 | 0.855927 |
| 11 | 0 | 0.0196098 | 8.0887e-019 | 0.0210538 | 0.0031707 | 0.907594 |
| 12 | 0 | 0.0167855 | 0.000250394 | 0.0195676 | 4.29418e-018 | 0.958812 |
| 13 | 0.0944849 | 0.0112657 | 0 | 2.36807e-018 | 0.0363442 | 0.908605 |
| 14 | 0.00692169 | 0.0121538 | 0.00207852 | 0.00215503 | 0.0315172 | 1 |
| 15 | 0 | 0.0158746 | 7.68761e-019 | 0.0177283 | 1.14697e-017 | 0.390023 |
| 16 | 0.0728352 | 0.00868438 | 1.75703e-019 | 0 | 0.0280174 | 0.252156 |
| 17 | 1 | 0 | 0 | 0 | 0.0625062 | 0.937594 |
| 18 | 0 | 0.0117559 | 0.000175325 | 0.0137047 | 2.01723e-014 | 0.959331 |

Table 4. Suppliers Efficiency

| Suppliers | Efficiency | Suppliers | Efficiency | Suppliers | Efficiency |
|-----------|------------|-----------|------------|-----------|------------|
| 1 | 1 | 7 | 0.909066 | 13 | 0.908605 |
| 2 | 0.833417 | 8 | 0.916572 | 14 | 1 |
| 3 | 0.744619 | 9 | 1 | 15 | 0.390023 |
| 4 | 0.525436 | 10 | 0.855927 | 16 | 0.252156 |
| 5 | 0.850091 | 11 | 0.907594 | 17 | 0.937594 |
| 6 | 0.81318 | 12 | 0.958812 | 18 | 0.959331 |



Fig. 1. Histogram of all suppliers with their efficiency score

For the current research conducted in 18 suppliers, the results are:

1. Suppliers 1, 9 and 14, the efficiency score is 1 so these suppliers are assumed to be 100% efficient.
2. Supplier 15 is probably the most inefficient in comparison to all other suppliers.
3. Suppliers 1, 9 and 14 would be the most suitable set of suppliers (or key suppliers).
4. By using this DE, the company can obtain a recommended combination of efficient suppliers.
5. Combination of supplier 1, 9, 14 would be the recommended supplier set while the company needing single-item suppliers
6. The results of the case indicate that the DE algorithm can solve the problem effectively.

7 Conclusions

Optimal supplier Selection is a challenging task among thousands of potential suppliers. The present study shows DE as a tool for selecting the optimal suppliers. In this study, we present a comprehensive green supplier selection framework. The first step is to construct a criterion set that containing both environmental, management aspects, which is suitable for real world applications. We than presents an approach to solve the multiple-criteria green supplier selection problem with the application of DE, for DEA. By using this approach, a company can obtain a recommended combination of efficient suppliers.

In this study, the goal was the application of DE to the efficient multi-criteria green supplier selection in the green SCM. The main motivation of this study was to gain an understanding of the mechanics of DE and to determine the accuracy of DE in generating the optimum solutions for the DEA based mathematical model, which is the underlying optimization problem for the aforementioned purchasing system.

Future research may investigate the use of the DE algorithm to solve more difficult problem such as multi objective supplier selection with considering environmental aspects.

References

1. Agarwal, P., Sahai, M., Mishra, V., Bag, M., Singh, V.: A review of multi-criteria techniques for supplier evaluation and selection. *International Journal of Industrial Engineering Computations* 2 (2011), doi:10.5267/j.ijiec.2011.06.004
2. Noci, G.: Designing green vendor rating systems for the assessment of a supplier's environmental performance. *European J. of Purchasing and Supply Management* 3(2), 103–114 (1997)
3. Humphreys, P.K., Wong, Y.K., Chan, F.T.S.: Integrating environmental criteria into the supplier selection process. *Journal of Material Processing Technology* 138, 349–356 (2003)
4. Selos, E., Laine, T.: The perceived usefulness of decision-making methods in procurement. In: *Seventeenth International Working Seminar on Production Economics*, Pre-prints, vol. 1, pp. 461–472 (2012)
5. 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)
6. Plagianakos, V., Tasoulis, D., Vrahatis, M.: A Review of Major Application Areas of Differential Evolution. In: Chakraborty, U.K. (ed.) *Advances in Differential Evolution*. SCI, vol. 143, pp. 197–238. Springer, Heidelberg (2008)
7. Wang, F., Jang, H.: Parameter estimation of a bio reaction model by hybrid differential evolution. In: *Proceedings of IEEE Congress on Evolutionary Computation (CEC 2000)*, pp. 410–417 (2000)
8. Joshi, R., Sanderson, A.: Minimal representation multi sensor fusion using differential evolution. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* 29(1), 63–76 (1999)
9. Ilonen, J., Kamarainen, J., Lampine, J.: Differential evolution training algorithm for feed-forward neural networks. *Neural Process. Lett.* 17(1), 93–105 (2003)
10. Ali, M., Siarry, P., Pant, M.: An efficient differential evolution based algorithm for solving multi-objective optimization. *European Journal of Operational Research* (2011)
11. Genovese, A., Koh, S.C.L., Bruno, G., Bruno, P.: Green supplier selection: a literature review and a critical perspective. Paper Presented at *IEEE 8th International Conference on Supply Chain Management and Information Systems (SCMIS)*, Hong Kong (2010)
12. Mazhar, M.I., Kara, S., Kaebnick, H.: Reusability assessment of components in consumer products—a statistical and condition monitoring data analysis strategy. In: *Fourth Australian Life Cycle Assessment Conference—Sustainability Measures For Decision Support*, Sydney, Australia (2005)
13. <http://www.london2012.com/documents/locog-publications/locog-guidelines-on-carbon-emissions-of-products-and-services.pdf> (accessed on May 12, 2013)
14. Dimitris, K.S., Lamprini, V.S., Yiannis, G.S.: Data envelopment analysis with nonlinear virtual inputs and outputs. *European Journal of Operational Research* 202, 604–613 (2009)
15. Ramanathan, R.: *An Introduction to Data Envelopment Analysis: A Tool for Performance Measurement*. Sage Publication Ltd., New Delhi (2003)
16. Srinivas, T.: Data envelopment analysis: models and extensions. *Production/Operation Management Decision Line*, 8–11 (2000)

17. Charnes, A., Cooper, W.W., Rhodes, E.: Measuring the efficiency of decision making units. *European Journal of Operational Research* 2(6), 429–444 (1978)
18. Banker, R.D., Charnes, A., Cooper, W.W.: Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Science* 30, 1078–1092 (1984)
19. Das, S., Abraham, A., Chakraborty, U., Konar, A.: Differential evolution using a neighborhood-based mutation operator. *IEEE Transaction on Evolutionary Computing* 13(3), 526–553 (2009)
20. Kumar, P., Mogha, S.K., Pant, M.: Differential Evolution for Data Envelopment Analysis. In: Deep, K., Nagar, A., Pant, M., Bansal, J.C. (eds.) *Proceedings of the International Conference on SocProS 2011*. AISC, vol. 130, pp. 311–320. Springer, Heidelberg (2012)
21. Jouni, L.: A constraint handling approach for differential evolution algorithm. In: *Proceeding of IEEE Congress on Evolutionary Computation (CEC 2002)*, pp. 1468–1473 (2002)
22. Coello, C.A.C.: Theoretical and numerical constraint handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering* 191(11-12), 1245–1287 (2002); *Differential Evolution for Data Envelopment Analysis* 319
23. Ray, T., Kang, T., Chye, S.K.: An evolutionary algorithm for constraint optimization. In: Whitley, D., Goldberg, D., Cantu-Paz, E., Spector, L., Parmee, I., Beyer, H.G. (eds.) *Proceeding of the Genetic and Evolutionary Computation Conference (GECCO 2000)*, pp. 771–777 (2000)

Comparison between Differential Evolution Algorithm and Particle Swarm Optimization for Market Clearing with Voltage Dependent Load Models

Deep Kiran, Bijaya Ketan Panigrahi, and A. R. Abhyankar

Department of Electrical Engineering, Indian institute of Technology Delhi, New Delhi-110016, India

{deepkiran,bkpanigrahi,abhyankar}@ee.iitd.ac.in

Abstract. The day ahead concept of real power market clearing is applied for generator scheduling in a deregulated environment. The earlier approach of cost minimization declines the consumer benefit instead considers the reduction of overall generation cost. This brings in limitation which is compensated by the approach of social welfare maximization which is to maximize the profit for the supplier and consumer i.e. societal benefit. During optimization of these objectives, the behavior of control variables are observed keeping the different technical and operating constraints of the system. The modeling of loads as voltage dependent is implemented so as to analyze the effect on load served maximization and voltage stability enhancement index. The former tries to serve the maximum loads and the later maintains the flat profile of the voltages to avoid the voltage collapse on overloading and fault situations. The obligation by market participants to achieve the best solution is done by differential evolution algorithm and particle swarm optimization. A comparison is also made between these two optimization techniques. The investigation is performed on IEEE 30 bus system.

Keywords: Social welfare maximization, Load served maximization, Voltage stability enhancement index, Load modeling, Differential evolution algorithm, Particle swarm optimization.

1 Introduction

The deregulation in electricity market has brought in many opportunities for the researchers to work on many tough areas which allow the market to work efficiently and to avoid thievery. Initially, the power system operator was only concerned with the generation cost minimization of the system which has got no complexity in it and it was easy to schedule the system as the complete system was owned by single entity, generally government undertakings, which makes it an inelastic system. This philosophy does not holds good for the age of industrial revolution where lot of big consumers are also interested in taking part

and have stake in the market. This brings in the concept of social welfare where the complete society is considered and each individual (supplier or consumer) can decide and pay the cost according to its buying or consuming capacity of electricity. However, it is difficult for now to consider each small individual rather group of small consumers i.e. distribution substation and big consumers can be part of the market. This makes the market more elastic and complex.

The above idea is jot down in an exquisite manner in [1] where a strategy for day ahead real power market clearing (DA-RPMC) is performed using genetic algorithm. All the operational and technical constraints are considered and results are observed. To attain the equilibrium in electric power exchange with unit commitment is discussed in [2] to maximize the profit of supplier side. In [3], all the complex bids are optimized together by considering security constraints to maximize the day ahead benefits. In [4], the elasticity of demand is discussed with bid cost minimization and social welfare maximization for the pool market. In [5], keeping the voltage stability as the prime concern, the maximization of the social welfare in a multi-objective environment. In [6], the different bids are optimized for different load scenario which shows the consumers capability who is not ready to pay the same amount for the per unit consumption.

To reach the best solution, a need of optimization technique which is robust, flexible, simple and reliable is required. There are many optimization techniques available in the literature which are quoted by different researchers. In [7], author have suggested a technique based on the genes which works on binary system. In [8], a concept on movement of flock of birds to search food is expressed mathematical which is an efficient technique for optimization. Authors of [9] have also shown the advantages and wide areas of research possible with this technique. In [10], an optimization technique based on the concept of using vector differences for perturbing the vector population which generates the trial vectors for the comparison.

The paper is organized as follows: in section 2, paper explains the utilization of objectives for DA-RPMC. Section 3 accounts for the use of optimization technique for achieving the best solution. Results are discussed in section 4 and conclusion of the paper is made in section 5.

2 Day Ahead Real Power Market Clearing (DA-RPMC)

The settlement for each unit (either supplier or consumer) is executed in DA-RPMC. Each player provide their complex bids in the day ahead market and gets scheduled for the next day. This avoids the allocation of major chunk of power at last moment. However, some small variations in schedule can be brought in the intra-day market which is not the motive of the paper. If the bids provided by the supplier itself is used for optimization makes the scenario as inelastic and if suppliers and consumers bids are considered for optimization than it will make the scenario as elastic. It has been assumed that the unit commitment schedules are already available and real power market is cleared.

2.1 Centralized Market with Inelastic Demands

In the inelastic demand scenario, as only suppliers take part in the market clearing, the optimization is performed only for generation cost minimization (CM) which is mathematically expressed as:

$$CM = \sum_{i=1}^{N_G} (\alpha_i P_{Gi} + \beta_i P_{Gi}^2) \quad (1)$$

where, α_i and β_i are cost coefficients of i^{th} generator and P_{Gi} is real power generation of i^{th} generator.

2.2 Centralized Market with Elastic Demand

In the elastic demand scenario, the generator and loads both take part in the market clearing. This provokes a situation where an optimization technique has to manage and maximize the surplus of generator and load such that they both enjoy the societal benefit. The maximization of this benefit is termed as social welfare maximization which is mathematically expresses as mentioned in [4,6]:

$$SWM = maximize \left(\sum_{i=1}^{N_D} B_{Di}(P_{Di}) - \sum_{i=1}^{N_G} C_{Gi}(P_{Gi}) \right)$$

where $B_{Di}(P_{Di}) = \sum_{i=1}^{N_D} (\gamma_i P_{Di} - \delta_i P_{Di}^2)$ (2)

$$C_{Gi}(P_{Gi}) = \sum_{i=1}^{N_G} (\alpha_i P_{Gi} + \beta_i P_{Gi}^2)$$

where γ_i and δ_i are demand coefficients of i^{th} load bus and P_{Gi} and P_{Di} is real power generation of i^{th} generator and real power demand of i^{th} demand. $C_{Gi}(P_{Gi})$ is the generation cost of real power P_{Gi} at bus i . $B_{Di}(P_{Di})$ is the gross consumer surplus for real power demand P_{Di} at bus i .

2.3 Load Modeling

The preponderate affect of voltage on the loads has made a point to consider it in the problem. The voltage dependency of loads are observed on many machines especially induction type which is most widely used in the recent appliances covering considerable portion of the loads. The effect includes the variation in torque, speed and prolonged heating which finally reduce its life. These needs to be modeled mathematically to be expressed in the market as mention in [11]:

$$P_L = P_{l0} \left(\frac{V}{V_0} \right)^{np} \quad (3)$$

$$Q_L = Q_{l0} \left(\frac{V}{V_0} \right)^{nq} \quad (4)$$

where, P_L and Q_L are the real and reactive power load, V_0 , P_{l0} and Q_{l0} are the nominal voltage magnitude, real and reactive power load respectively. np and nq are the voltage exponent which dependent on the type and composition of the load. These equations depicts the real time dynamics of the loads with the variation in the voltage profile of the load buses.

2.4 Load Served Maximization (LSM)

The LSM is the one of the objectives which is also implemented in the system whenever the loads are modeled as voltage dependent. A mandatory check is required to watch for the loads behavior and its divergence from its original value due to voltage dependency. The optimization of CM and SWM affects the load served. This makes it necessary to be used as the supplementary function and prevent the extent of load variation. The mathematical expression is:

$$LSM = \sum_{i=1}^n P_{l0}^i \left(\frac{V_i}{V_{0i}} \right)^{np} \quad (5)$$

2.5 Voltage Stability Enhancement Index (VSEI)

An index termed as L-index was established in [12] which is evaluated to understand the voltage stability margin. It provides the value which suggests the stretch to violate stability limit. This value is the result of load flow solution. It varies between 0 (a no load scenario) and 1 (a voltage collapse scenario). Minimization of VSEI maximizes the system capability to push the voltages up. The mathematical expression is:

$$L_j = \left| 1 - \sum_{i=1}^{ng} F_{ji} \frac{E_i}{E_j} \right| \quad (6)$$

$$L - index = VSEI = \sum_{i=ng+1}^n L_j^2 \quad (7)$$

where, n is number of buses, ng is number of generators in the system. E_i , E_j are the complex voltages of generator and load buses respectively. The values of F_{ji} are obtained from the Y-Bus matrix. It will be good to observe VSEI deflections as other objectives optimizes.

2.6 Equality and Inequality Constraints of DA-RPMC

The equality and inequality constraints for the above problem are as follows:

Nodal Power Balance Constraints. The power balance includes real and reactive power balances

$$P_{Gi} - P_{Di} = \sum_j (V_i V_j Y_{ij} \cos(\theta_{ij} + \delta_j - \delta_i)) \tag{8}$$

$$Q_{Gi} - Q_{Di} = - \sum_j (V_i V_j Y_{ij} \sin(\theta_{ij} + \delta_j - \delta_i)) \tag{9}$$

Generation Capacity Limits. The real power generations are kept constant. The reactive power generations can vary as

$$Q_{Gi}^{min} \leq Q_{Gi}; (P_{Gi}^2 + Q_{Gi}^2)^{1/2} \leq S_{Gi}^{max} \tag{10}$$

Shunts Limits. The shunts have restrictions as follows

$$Q_{Ci}^{min} \leq Q_{Ci} \leq Q_{Ci}^{max} \tag{11}$$

Security Constraints. The security related constraints are as follows

$$V_i^{min} \leq V_i \leq V_i^{max} \tag{12}$$

$$T_i^{min} \leq T_i \leq T_i^{max} \tag{13}$$

where, i and j are the bus numbers, ij represents the line between bus i and j , P_G and P_D are the real power generation and demand in Mega-Watts (MW), Q_G and Q_D are the reactive power generation and demand in Mega-VAr (MVA), S_G is the MVA generated by the generator, V and δ are the voltage magnitude in per unit and angle in radians, Y and θ are the admittance magnitude in per unit and angle in radians respectively, T is the transformer tap settings, superscript min and max represents the minimum and maximum limits of the quantity respectively.

3 Optimization

The need for the best solution is required in every field of engineering and its omnipresent. Since the inception of the applications of optimization techniques has brought in gradual complexities in the problem. The recent world problems are most complex due to keeping consideration of every equality and inequality constraints. This is being possible because computational advancement in recent years. Many optimization techniques have also been developed and has worked exceptional in large and complex problems. With respect to our problem which is a continuously varying environment because of voltage dependency of loads. As researchers have claimed the high performance of differential evolution algorithm and particle swarm optimization has proved its presence in the applications with sufficiently steady, agile and precise in nature. With these qualities, adoption and implementation to our problem was simpler and efficient to achieve the optimal solution among the complete search space.

3.1 Differential Evolution (DE)

DE being one of the fast and reliable optimization technique which has, in general, better performance than the previous quoted techniques in literature. General process for any evolutionary algorithm is satisfied and followed by DE. The algorithm is mentioned in [10,13].

Nevertheless, contrary to classical evolutionary algorithm, it explores the search space by taking differences of the parameter vectors. In its novice state, DE accelerates the each randomly generated vector with the mismatch between tertiary randomly generated vector and itself to generate a donor vector, also termed as target vector, for that randomly generated vector. Then the crossover operation is performed on the target vector and donor vector which will create a trial vector. This trial vector has to compete with the parent vector of the same index, which was initially generated, to perform the selection of the fittest among all. The winners are now the new parent vectors which is again optimized for the next generation until the stopping criterion has reached. The idea is expressed in the flowchart mentioned in Fig. 1 which conveys the concept of adaptive DE by changing the mutation operators with the increase in the number of iterations.

3.2 Particle Swarm Optimization (PSO)

PSO is another acclaimed optimization technique which has the ability to learn from individuals past mistakes and follow the best individual of the group. The algorithm is mentioned in [8,15].

PSO also works on the general procedure of evolutionary algorithms. The randomly generated particles are trained to learn from its past and from the overall best particle of that set. Each of the particles moves with some velocity towards the best solution obtained in each iteration. The evaluation of this velocity needs three components: particles own weighted velocity, a factor of the mismatch between the particles self best and its present location and a factor of the difference between the particles social best and its present location. These components are augmented to push the particles towards the best solution achieved so far. The new particles are improvised until the stopping criterion has reached.

3.3 Fitness Function

The objective functions changes but the forfeiture of the breached vectors remains the same. The penalty is added for the bus voltages minimum & maximum limits, reactive power minimum & maximum limits and line limits. The fitness function can be formulated as

$$Fitness\ function = \frac{1}{1 + F_{aug}} \quad (14)$$

$$F_{aug} = F + \lambda((V - V_{max})^2 + (V - V_{min})^2 + (Q - Q_{min})^2 + (S - S_{max})^2) \quad (15)$$

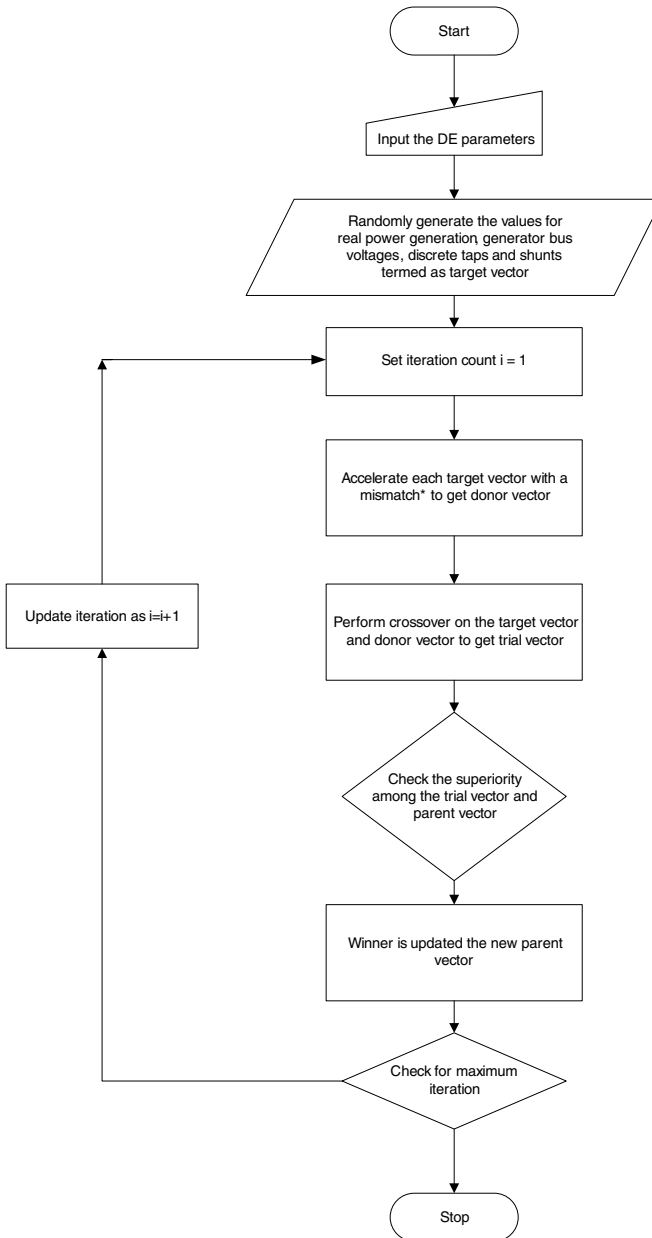


Fig. 1. Flowchart of differential evolution algorithm. mismatch* is evaluated by different mutation operators mentioned in [14]

where, F_{aug} is the power system augmented operating objective, λ is penalty multiplier, V_{max} and V_{min} is maximum and minimum voltage limit on each bus, Q_{max} and Q_{min} is maximum and minimum reactive power limit on each bus respectively, S_{max} is the maximum MVA injection limit by the generator.

This fitness function is judiciously implemented for all the objectives discussed above with a proper sense of minimization and maximization type of problem. Although, the soft constrained penalty imposed on the objective function remains the same to restrict the violation of the technical and operational parameters.

4 Results and Discussion

The optimization of the above objectives are performed on MATLAB. The investigation is performed on standard IEEE 30 bus system (data given in [16]). This is a six unit system having 21 loads and 41 branches in which four branches are having transformer taps. Once the bids (given in appendix) are received by the system operator from all the suppliers and consumers then system operator will run a joint optimization program considering all the operational and technical constraints for DA-RPMC scheduling. The optimality of the solution is reached by considering five real power generation, six generator voltages, four transformer taps and nine shunt reactors as the control variables in the system. The discrete transformer taps ranging from 0.9 pu to 1.1 pu with a step of 0.01 pu and the discrete shunt reactors has been varied between 0 pu to 5 pu with a step of 1 pu. The exponential load is modeled as $np=1$ and $nq=2$. The upper and lower limit of generator buses are 1.10 pu and 0.95 pu respectively. Similarly, the upper and lower limit of load buses are 1.05 pu and 0.90 pu respectively.

To observe the effect of voltage dependent loads, DA-RPMC is evaluated for each objective function individually. To make the analysis more stern, the examination of voltage collapse is also performed during a period of overload and faults which is a stressed condition. The overloading effect is addressed by increasing the loads by 30% to 368.42 MW from its nominal load of 283.4 MW. The fault is created by violating the line between the buses 27 and 28.

4.1 DA-RPMC Scheduling

The analysis is performed for inelastic and elastic demand scenarios. As explained earlier, in the case of inelastic demand, the consumers will continue to absorb power from the connecting bus irrespective of the price variations of per unit consumption. On the contrary, in the case of elastic demand, the consumers vary their power consumption according to the change in the per unit price of consumption. These effects are analyzed and compared on the basis of fitness function.

Cost Minimization (CM). The single objective values are optimized by DE and PSO for the inelastic and elastic demand scenario are presented in Table

1. It is observed from both the conditions that CM reduces the generation cost by reducing the voltage dependent loads. DE has reduced the generation cost to 741.6653 \$/hr in compared to 753.6438 \$/hr of PSO. This increase in cost is due to not reduction in voltages on which has increased the loads served as well. This increment in cost has also reduced the societal benefits enjoyed by every party. The decrease in the social welfare from unstressed condition to stressed condition is due to the overloading which has increased the generation cost and reduces the social welfare.

Table 1. CM with DE and PSO under unstressed and stressed condition

| Function | DE | | PSO | |
|-------------------------|-----------------|----------------|-----------------|------------------|
| | Unstressed | Stressed | Unstressed | Stressed |
| Generation cost (\$/hr) | 741.6653 | 993.019 | 753.6438 | 1041.9882 |
| Load served (MW) | 263.4585 | 321.473 | 268.1993 | 340.8474 |
| VSEI | 0.1436 | 0.3564 | 0.1343 | 0.7565 |
| Social welfare (\$/hr) | 45251.7039 | 22312.4642 | 45134.9527 | 30011.3073 |

The minimization of VSEI is to check for voltage stability which is adequately within safety limits of 0.1436 for unstressed condition of DE and 0.1343 from PSO. However, overloading and violation in the line, in the case of DE, has seen a sudden increase in its value to 0.3564 which is still within limits whereas, in the case of PSO, it has seen an acute increment which needs to be taken care of.

Social Welfare Maximization (SWM). The DE and PSO is implemented to obtain the objective values for the unstressed and stressed condition and presented in the Table 2. The SWM with DE has attained a value of 51415.6645 \$/hr which is maximum among optimization of all objectives for unstressed condition whereas, for PSO, social welfare has reached maximum value of 50243.6235 \$/hr. It is visible that the generation cost has increased compared from the case of CM (Table 1). It is also seen that voltage dependent loads have not reduced much from its nominal load whereas, in CM case, the loads have reduced to much more value. This means that the bus voltages are maintained at near 1 pu or flat profile. Similar observations can also be made in stressed condition where a mere load reduction is seen from its overloaded value.

An enormous increase has been seen in the minimization of VSEI, with DE, from 0.1758, of unstressed condition which is near stability, to 0.9613, of stressed condition which is an alarming situation as it approaches voltage collapse for any further increase in the loading. Optimization through PSO has also increased the VSEI similarly due to the efforts of social welfare to provide equal benefits to suppliers and consumers.

Load Served Maximization (LSM). Table 3 represents the single objective values of all the functions under unstressed and stressed condition with DE and

Table 2. SWM with DE and PSO under unstressed and stressed condition

| Function | DE | | PSO | |
|-------------------------|-------------------|-------------------|-------------------|-------------------|
| | Unstressed | Stressed | Unstressed | Stressed |
| Generation cost (\$/hr) | 852.5605 | 1138.7735 | 792.6715 | 1186.0809 |
| Load served (MW) | 279.0418 | 360.8517 | 274.9044 | 359.6398 |
| VSEI | 0.1758 | 0.9613 | 0.1702 | 0.9385 |
| Social welfare (\$/hr) | 51415.6645 | 37874.8499 | 50243.6235 | 37412.7107 |

PSO. The voltage of the buses are pushed towards its upper limits to increase the loads to its maximum of 299.5021 MW and 293.5914 MW which has increased the generation cost for unstressed condition with DE and PSO respectively. This increase in the generation cost has reduced the social welfare for both conditions which is the least value obtained among other objective optimization which portrays the benefits enjoyed by one party (either supplier or consumer) over the loss of other party. The stressed condition shows not much difference while optimization with DE and PSO which has increased the loads to 379.4563 MW and 378.8983 MW respectively.

Table 3. LSM with DE and PSO under unstressed and stressed condition

| Function | DE | | PSO | |
|-------------------------|-----------------|-----------------|-----------------|-----------------|
| | Unstressed | Stressed | Unstressed | Stressed |
| Generation cost (\$/hr) | 890.5848 | 1213.1057 | 871.7076 | 1222.5814 |
| Load served (MW) | 299.5021 | 379.4563 | 293.5914 | 378.8983 |
| VSEI | 0.1982 | 0.8613 | 0.1887 | 0.9598 |
| Social welfare (\$/hr) | 40798.4972 | 17610.9794 | 41407.0871 | 18135.5982 |

The VSEI value, a dimensionless quantity, shows the gap between the generator bus voltages and load bus voltages. This gap has increased from 0.1982 (unstressed condition) to 0.8613 (stressed condition) with DE and from 0.1887 (unstressed condition) to 0.9598 (stressed condition) with PSO which is a steep ascend and needs to be avoided so as to evade the voltage collapse situation.

Voltage Stability Enhancement Index Minimization (VSEIM). The objective values obtained with DE and PSO for all the functions and for both the conditions are presented in Table 4. The minimization of VSEI has obtained a value of 0.1220 with DE and 0.1262 with PSO which shows an almost flat profile situation not necessarily 1 pu rather voltages are reduced all around the system which reduces the generation cost and the load served for unstressed and stressed condition. Social welfare also provides an inferior values compared to SWM case (Table 2). A similar trend is seen in the stressed condition where VSEI has obtained a minimum value of 0.5369 with DE and 0.5786 with PSO

Table 4. VSEIM with DE and PSO under unstressed and stressed condition

| Function | DE | | PSO | |
|-------------------------|--------------|---------------|---------------|---------------|
| | Unstressed | Stressed | Unstressed | Stressed |
| Generation cost (\$/hr) | 757.4282 | 1010.1643 | 883.9927 | 1088.9904 |
| Load served (MW) | 264.9155 | 330.7677 | 274.5731 | 340.7693 |
| VSEI | 0.122 | 0.5369 | 0.1262 | 0.5786 |
| Social welfare (\$/hr) | 45944.7523 | 26183.9339 | 47206.6683 | 26609.9824 |

which is in the midway of stability and collapse of voltage. It is understandable that the generation cost, load served and social welfare obtained is the second best among other objective optimization.

5 Conclusion

The investigation of DA-RPMC on IEEE 30 bus test system mimics the actual functioning of centralized market operated by system operator. The scheduling of generators and assigning the required values for generator bus voltages, discrete transformer taps and discrete shunts are the control variables in the system. The system operator, in its joint optimization algorithm, keeps a regular check on the operating and technical constraints of the system and reject the solution for any type of violation. The cost minimization and social welfare maximization are implemented for scheduling. Both the objective shows a peculiar characteristics which is addressed in this paper. The inelastic demand situation or CM only tries to reduce the overall generation cost whereas the elastic demand situation or SWM tries to provide same privilege to both supplier and consumer. The analysis has enlightened the understanding of SWM which maintains the loads at near its nominal value by not varying the bus voltages much.

The adoption of load served maximization as one of the objectives has shown the bus voltages increment to maximize the total loads fed by the system which increases the generation cost and reduces the social welfare from its optimum value. Voltage stability enhancement index has played a vital role in the optimization of these objectives by providing the check for voltage stability. The increase in the gap between generator bus voltages and load bus voltages increases the chance of voltage collapse by increasing the loading of the system.

The two conditions (unstressed and stressed) has showed the worst possibility of system to disintegrate. The analysis of the system at stressed condition is performed to observe the maximum overloading capability and to bear fault situations. It brings the system at emergency state where voltages are at the verge of collapse if any further overloading or fault takes place.

The optimization of the above objectives to achieve the most optimum solution is done by DE and PSO. Being among the most robust, simple, efficient, precise and fast optimization tool available in the literature has convinced to be implemented in this type of problem. The increased intricacy of the optimized solution which has different real and reactive power demands due to the

implementation of voltage dependent load models. This has restricted the best solution obtained to be repeated with some modification. Instead of all these complications, the solutions obtained by DE are much better than PSO. This has been only possible due to the exploitative and exploring capabilities of adaptive DE to hunt for the complete search space. The results obtained have also proved the dominance.

A Bid Data

The generation cost coefficients and demand cost coefficients are given in Table 5 and 6 respectively.

Table 5. Generation cost coefficients

| Bus | α | β | Bus | α | β |
|-----|----------|---------|-----|----------|---------|
| 1 | 2 | 0.00375 | 8 | 3.25 | 0.00834 |
| 2 | 1.75 | 0.0175 | 11 | 3 | 0.025 |
| 5 | 1 | 0.0625 | 13 | 3 | 0.025 |

Table 6. Demand cost coefficients

| Bus | γ | δ | Bus | γ | δ |
|-----|----------|----------|-----|----------|----------|
| 1 | 0 | 0 | 16 | 591.4281 | 8.2983 |
| 2 | 928.7555 | 8.578 | 17 | 495.8965 | 7.4525 |
| 3 | 147.8876 | 7.0669 | 18 | 830.7741 | 2.2425 |
| 4 | 761.5075 | 9.3577 | 19 | 539.6555 | 3.7352 |
| 5 | 515.227 | 1.2934 | 20 | 870.7727 | 9.798 |
| 6 | 0 | 0 | 21 | 369.3071 | 3.7753 |
| 7 | 524.5606 | 7.3147 | 22 | 0 | 0 |
| 8 | 122.0792 | 8.4407 | 23 | 154.6799 | 7.5734 |
| 9 | 0 | 0 | 24 | 716.3914 | 6.8708 |
| 10 | 394.9041 | 2.9905 | 25 | 445.1295 | 6.992 |
| 11 | 0 | 0 | 26 | 920.9361 | 9.9797 |
| 12 | 890.885 | 8.1691 | 27 | 0 | 0 |
| 13 | 0 | 0 | 28 | 0 | 0 |
| 14 | 934.6399 | 9.0153 | 29 | 936.2426 | 7.3685 |
| 15 | 399.4201 | 4.1416 | 30 | 238.2213 | 9.6713 |

References

1. SurenderReddy, S., Abhyankar, A.R., Bijwe, P.R.: Multi-objective day-ahead real power market clearing with voltage dependent load models. *International Journal of Emerging Electric Power Systems* 12(4) (2011)
2. Motto, A., Galiana, F.: Equilibrium of auction markets with unit commitment: the need for augmented pricing. *IEEE Transactions on Power Systems* 17(3), 798–805 (2002)
3. Madrigal, M., Quintana, V.: Optimal day-ahead network-constrained power system's market operations planning using an interior point method. In: *IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 1, pp. 401–404 (1998)
4. Wu, J., Guan, X., Gao, F., Sun, G.: Social welfare maximization auction for electricity markets with elastic demand. In: *7th World Congress on Intelligent Control and Automation*, pp. 7157–7162 (2008)
5. Milano, F., Canizares, C., Invernizzi, M.: Multiobjective optimization for pricing system security in electricity markets. *IEEE Transactions on Power Systems* 18(2), 596–604 (2003)
6. Liu, H., Shen, Y., Zabinsky, Z., Liu, C.C., Courts, A., Joo, S.K.: Social welfare maximization in transmission enhancement considering network congestion. *IEEE Transactions on Power Systems* 23(3), 1105–1114 (2008)
7. Goldberg, D.E., Holland, J.H.: Genetic algorithms and machine learning. *Machine Learning* 3(2), 95–99 (1988)
8. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948 (1995)
9. AlRashidi, M.R., El-Hawary, M.: A survey of particle swarm optimization applications in electric power systems. *IEEE Transactions on Evolutionary Computation* 13(4), 913–918 (2009)
10. 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)
11. Kundur, P.: *Power System Stability and Control*. McGraw-Hill, Inc., New York (1994)
12. Kessel, P., Glavitsch, H.: Estimating the voltage stability of a power system. *IEEE Transactions on Power Delivery* 1(3), 346–354 (1986)
13. Das, S., Suganthan, P.: Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation* 15(1), 4–31 (2011)
14. Qin, A.K., Suganthan, P.: Self-adaptive differential evolution algorithm for numerical optimization. In: *The IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1785–1791 (2005)
15. Del Valle, Y., Venayagamoorthy, G., Mohagheghi, S., Hernandez, J.C., Harley, R.: Particle swarm optimization: Basic concepts, variants and applications in power systems. *IEEE Transactions on Evolutionary Computation* 12(2), 171–195 (2008)
16. Alsac, O., Stott, B.: Optimal load flow with steady-state security. *IEEE Transactions on Power Apparatus and Systems* PAS-93(3), 745–751 (1974)

Multipopulation-Based Differential Evolution with Speciation-Based Response to Dynamic Environments

Souvik Kundu, Debabrota Basu, and Sheli Sinha Chaudhuri

Dept. of Electronics and Telecommunication Engg., Jadavpur University, Kol-32, India
{sk210892, basudebabrota29}@gmail.com

Abstract. Unlike static optimization problems, the position, height and width of the peaks may vary with time instances in dynamic optimization problems (DOPs). Many real world problems are dynamic in nature. Evolutionary Algorithms (EAs) have been considered to solve the DOPs in the recent years. This article proposes a multi-population based Differential Evolution algorithm which uses a local mutation to control the perturbation of individuals and also avoid premature convergence. An exclusion rule is used to maintain the diversity in a subpopulation to cover a larger search space. Speciation-based memory archive has been used to utilize the previously found optimal information in the new change instance. Furthermore the proposed algorithm has been compared with four other state-of-the-art EAs over the Moving Peak Benchmark (MPB) problem and a benchmarks set named Generalized Dynamic Benchmark Generator (GDBG) proposed for the 2009 IEEE Congress on Evolutionary Computation (CEC) competition.

Keywords: Differential Evolution, local mutation, multi-population, dynamic optimization problems, speciation.

1 Introduction

Differential Evolution (DE) [1] is a very simple and popular algorithm for solving global optimization problems. It operates by means of computational steps which are similar to the EAs. However, unlike EAs, the members are perturbed by the scaled differences of the randomly taken and distinct vectors from the whole population. As it does not require any separate probability distribution, it is implicitly adaptive in this aspect. The popularity of DE is proliferating due to its simple structure, compactness, robustness and the parallel searching mechanism.

Many real world problems are dynamic in nature. In case of DOPs, optimal solutions change with time. Hence we require algorithms that can detect the change in environment and should be able to track the optimum continuously [2]. A few dynamic real world problems are: price fluctuations, machine breakdown or maintenance, financial variations, stochastic arrival of new tasks etc. The main drawback of the conventional EAs under dynamic environment is loss of diversity,

i.e. premature convergence in a local peak which makes it unable to track the global optimum. So, in case of DOPs the main challenge is to maintain the diversity and as well as to produce high quality solutions by tracking the moving optima.

Classical DE faces difficulties when applied to the DOPs. There is a tendency of DE individuals to converge prematurely into local optima [3, 4]. Hence, if a change in the environment occurs DE lacks the sufficient explorative power to detect the new optimum. Sometimes DE stops generating new optimal solution even without being trapped in local optima due to the *stagnation* effect. Even after the arrival of new solutions in the population, the algorithm does not progress towards optimal solution better than its current best. Researchers have developed several DE variants for tackling the difficulty of the dynamic optimization problems.

In this article the authors have proposed a multi-population based DE where mutation is performed using the individuals that belong to a particular subpopulation. This local mutation helps to avoid the premature convergence of the population individuals and also maintain the fine searching ability. An exclusion rule is also used to maintain the diversity in a particular subpopulation thereby covering a wide search range in the problem landscape. A *test individual* is used to detect the change in environment. At every generation fitness of this test individual is evaluated. If the fitness in a current generation does not match with the previous one, we can say that a change in environment occurred. To adapt into the new environment, a speciation based memory archive has been used that contains better solutions from the previous environment. From now on, our algorithm will be called as MLDES (multipopulation-based locally mutated differential evolution with speciation archive).

The rest of the paper is organized as follows: Section 2 contains the basic DE framework and overview of related works in EAs applied in DOPs is given. The structural components of our proposed algorithm MLDES is briefly described in Section 3. Experimental setup is discussed in Section 4 and the results obtained on the considered benchmarks are provided in Section 5. The working mechanism of our proposed algorithm is elucidated in Section 6. The paper is concluded in Section 7.

2 Scientific background and Related Works

2.1 Differential Evolution Framework

The basic DE framework proposed by Storn and Price [1] can be outlined in the following steps—

1) Initialization

The preliminary step involves the creation of NP members that represent the tentative solution to an objective function. As the iteration proceeds, these D -dimensional vectors (population members) represented by-

$$\overline{X}_i^G = \{x_{i,1}^G, x_{i,2}^G, x_{i,3}^G, \dots, x_{i,D}^G\} \quad \forall i \in \{1, 2, 3, \dots, NP\} \quad \text{where } G \in \{0, 1, \dots, G_{\max}\}$$

are improved upon its existing values.

The initial trial solutions can be represented as-

$$x_{i,j}^0 = x_{\min,j} + rand \times (x_{\max,j} - x_{\min,j}) \quad \forall j \in [1, D] \quad (1)$$

where *rand* is a random number sampled independently from uniform distribution in the range [0,1] and $x_{\min,j}$ and $x_{\max,j}$ are the lower and upper bound of j^{th} dimension.

2) Mutant Vector Generation: Mutation

The mutation operator, inspired from the dynamics of gene-tic materials in living organisms, which propels the members towards different regions of search space by creation of new individuals from a stochastically selected sample of the population. Given below are some mutation techniques used in this paper for integrating with our approach:

1 DE/rand/1— $v_i^G = x_{r1}^G + F \times (x_{r2}^G - x_{r3}^G)$

(2)

2 DE/best/1— $v_i^G = x_{best}^G + F \times (x_{r1}^G - x_{r2}^G)$

(3)

3 DE/current-to-best/1— $v_i^G = x_i^G + F \times (x_{best}^G - x_i^G + x_{r1}^G - x_{r2}^G)$ (4)

4 DE/current-to-rand/1— $v_i^G = x_i^G + F \times (x_{r1}^G - x_i^G) + F \times (x_{r2}^G - x_{r3}^G)$ (5)

5 DE/rand/2— $v_i^G = x_{r1}^G + F \times (x_{r2}^G - x_{r3}^G) + F \times (x_{r4}^G - x_{r5}^G)$ (6)

In the above notations, the indices $r1, r2, r3, r4$ and $r5$ are mutually exclusive members with respect to the base vector. The scaling factor parameter F controls the degree of positional modification. Literature proposes that it should be limited to the range [0, 2]. Careful tuning of F is a prerequisite for successful optimization of a problem and it is advisable to limit its value in the range [0.4, 0.95].

3) Crossover: Generation of Offspring Vector

Recombination of genetic materials ensures diversity in a population. This is mapped in DE through crossover operator. Two predominant modes are—*binomial* and *exponential* crossover. The binomial crossover is used in our work and in this model a uniform number sampled independently in the range (0, 1) and is compared against the value of control parameter crossover rate CR . If it is less than CR , new genetic material from the mutant individual is inserted in the corresponding component of the *offspring* vector U otherwise the parent gene is included. The method is given below.

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } rand \leq Cr \text{ or } j = jrand \\ x_{j,i,G} & \text{otherwise} \end{cases} \quad (7)$$

The crossover rate CR also affects the performance of DE. A high value is preferred for functions exhibiting variable link-ages while a low value is ideal for separable functions.

4) Fitness-proportionate Selection

The natural process of selection that ensures the fittest members of the population survive after biological evolution is simulated by a fitness proportionate selection between the new offspring and the parent individual. The less fit individual is replaced by the fitter counterpart. The selection process is carried out as (minimization)—

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G} & \text{if } f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G}) \\ \vec{X}_{i,G} & \text{if } f(\vec{U}_{i,G}) > f(\vec{X}_{i,G}) \end{cases} \quad (8)$$

2.2 EAs for DOPs – A Brief Overview

In 1966, the first attempt was taken for solving DOPs using EAs [5]. Since late 1980s, it started to attract several researchers and results a huge increase in the number of publications in this area. Comprehensive surveys on the adaptation and application of EAs for tackling DOPs can be found in [2, 7, 8]. A number of articles to solve DOPs using Genetic Algorithm (GA) have published by the researchers [8-10]. Besides GAs, swarm-intelligent algorithms like particle swarm optimization (PSO) and artificial bee colony (ABC) algorithms have also earned a lot of interest in solving DOPs [12-14]. Since late 1990s, DE [15-18] has started to get attention from DOP researchers due to its parallel searching procedure, and the capability of providing better exploration-exploitation trade off. Mendes and Mohais proposed DynDE [16]. Angeria and Santosh presented Trigonometric Differential Evolution (TDE) [17]. Many researchers have also tried to hybridize DE with other state-of-the-art EAs. Lung and Dumitrescu proposed CESO [18], where two collaborative populations were used; one evolves according to DE and other evolves with a PSO algorithm.

3 MLDES Algorithm

In this section the proposed algorithm has been described along with its all salient features.

3.1 Local Mutation

The population is initialized with a structure of different subpopulations to maintain a larger diversity. Now in a particular subpopulation, mutation phase is performed. In this paper we have introduced a new mutation scheme to enhance the diversity throughout the population and also generate quality solutions.

The mutant vector is formed by a base vector, a different vector and a unit vector formed by the Gaussian random number. Base vector is the current individual here. The difference vector is formed by the neighborhood concept. Here we have taken the difference between the current individual and best individual in the neighborhood with respect to the current individual. In this case the scaling factor is varied adaptively. During the generation process of the mutant vector the scale factor is independently

generated for each dimension of each differential vector. The unit vector part will be formed by the Gaussian random number. Scaling factor for this part will be determined by the average difference of the dimension wise gap between the current individual and the best individual of the subpopulation where the current individual exists.

The mutation scheme can be expressed as:

$$\vec{V}_{mut}^G = \vec{X}_i^G + F \times (\vec{X}_{best}^G - \vec{X}_i^G) + d \times \hat{N} \tag{9}$$

Here \vec{X}_i^G is the current individual of G^{th} generation. \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 $(f(\vec{X}_k^G)/f(\vec{X}_i^G)-1)/R_{ik}$ ($k = 1, 2, \dots, M$ where $M =$ Number of individuals in the subpopulation and $k \neq i$) is maximum. Here R_{ik} is

the Euclidean distance between \vec{X}_i^G and \vec{X}_k^G . In this article we aim at elevating F whenever a particle is situated far away from the favourable region where the suspected optima lies, i.e. the fitness value of the particle differs much from the best solution value. On the other hand we should reduce F whenever the objective function value of the particle nears the best solution. These particles are subjected to lesser perturbation so that they can finely search the surroundings of some suspected optima. The scheme may be outlined as:

$$F_i = F_{min} + (F_{max} - F_{min}) \cdot \frac{\Delta f_i}{1 + \Delta f_i} \tag{10}$$

where F_i represents the value of the scale factor for the i^{th} target vector and $\Delta f_i = |f(\vec{X}_i) - f(\vec{X}_{best})|$. F_{min} and F_{max} are the lower and upper bounds of F set to 0.2 and 0.9 respectively. As clear from equation (10), F_i depends on the factor $\frac{\Delta f_i}{1 + \Delta f_i}$. The factor $\frac{\Delta f_i}{1 + \Delta f_i}$ can be modified to $(1 + 1/\Delta f_i)^{-1}$. For a target vector if Δf_i is large $1/\Delta f_i$ decreases, so the factor $(1 + 1/\Delta f_i)^{-1}$ increases, so F_i gets enhanced and the particle will be subjected to larger perturbation so that it can jump to a favourable region in the landscape. For the best individual $\Delta f_i = 0$, so F_i will be equal to F_{min} (0.2) which is evident because the best vector is required to undergo a lesser perturbation so that it can perform a fine search within a small neighbourhood of the suspected optima. Thus the particles which are distributed away from the current fittest individual have large F_i values and keeps on exploring the search space, maintaining sufficient diversity.

In the second part \hat{N} is the unit vector generated by Gaussian random number with zero mean and unity variance. Let a vector \vec{N} formed by the Gaussian random numbers. Then $\hat{N} = \vec{N}/\|\vec{N}\|$ where $\|\vec{N}\|$ is the magnitude of \vec{N} . Here the scaling

factor, $d = \sum_{j=1}^D (\bar{L}_{best,j} - \bar{X}_{i,j}) / D$. $\bar{L}_{best,j}$ is the j^{th} dimension of the best vector in the corresponding subpopulation. D is the dimensionality of the problem.

Here, due to the normal random number generated unit vector, every individual will be perturbed in a random direction by an amount of scale parameter named d . For fusion of the two parts in the mutation process a better exploration exploitation trade off has been obtained.

3.2 Exclusion Principle

In DOPs, it is very important that the population members are evenly distributed in the search space. Otherwise it will lose exploration capability. In order to maintain the population diversity we have developed an exclusion principle to maintain the population individuals at different basins of attraction.

The Euclidean distance between the best individuals of all the subpopulations are calculated at each generation. If the distance between a pair of subpopulation best individuals lies below a certain threshold then the whole subpopulation, which contain comparatively worse best individual, is reinitialized. Now the threshold is calculated according to the following rule:

Let the search range is Z and the problem is D dimensional. There are SP numbers of subpopulations. Then the *threshold* will be calculated as: $threshold \leq Z / (SP * D)$.

3.3 Environment Change Detection

An efficient dynamic optimizer must detect the change of environment effectively. Here in this algorithm change detection in environment can be done by a *test solution*. At the very beginning of the algorithm a *test solution* is placed which does not take part in the optimization process. But at every generation the fitness value of the *test solution* is calculated. Whenever there is a change in the fitness value from the previous generation, it can be said that a change of environment has taken place and the algorithm needs to take necessary actions.

3.4 Speciation-Based Memory Archive

In most of the dynamic optimization problems the new environment is similar to the old one to a certain extent. So it is very natural that an algorithm will perform better after detecting an environment change if good solutions can be preserved in a memory archive. When change detection occurs the proposed algorithm does not reinitialize the entire population. Some of the population members are conserved for the following environment. Hence for storage of the individuals, the whole population is partitioned into a number of species. In each species there must be a species seed. Now let us consider there is m number of individuals in each of the species with a species seed which provide maximum fitness value within a stipulated species. No species radius is taken for partition. Rather the nearest neighborhood concept is used. For simplicity we

have taken m same as *subpopulation number*. In each of the species, the species seed and half of the individuals taken randomly are left unchanged in the archive for the following environment and the rest are reinitialized randomly in the search space. The combination of half newly generated member and half species-based individuals from previous environment constitutes the new population. The detailed algorithmic construction of the speciation-based memory archive can be given by *Algorithm 2*.

The pseudo code of the entire algorithm is given in *Algorithm 3*.

Algorithm 2. Speciation-Based Memory Archive: Reinitialization Of Population After A Change In Environment

Input: Population of the previous environment

- 1 Sort all the individuals in the population in the ascending order of their fitness values.
- 2 **for** $i = 1: no_of_species$
- 3 Set the best unprocessed individual as species seed.
- 4 Find the nearest m individuals of the species seed and set them as one species corresponding to the above mentioned species seed.
- 5 Remove the processed members from the current population.
- 6 Randomly pick $m/2$ individuals and reinitialize them within the search space.
- 7

4 Experimental Setup

For experimentation we have considered two sets of benchmark problems.

4.1 Moving Peaks Benchmark (MPB) Problem

Moving Peak Benchmark (MPB) problem proposed by Branke [19]. It is a widely known in dynamic optimization problem (DOP). The MPB problem allows us to alter the location, height and width of peaks. The default configuration is given in Table 1.

4.2 Generalised Dynamic Benchmark Generator (GDBG)

CEC 2009 benchmark problems for dynamic optimization use the generalized dynamic benchmark generator (GDBG) proposed in [20] that constructs dynamic environments for the location, height and width of the peaks. Instead of the shifting method as in the MPB, a rotation method was introduced in it. The GDBG system poses a greater challenge due to this rotation method, larger number of local optima and the higher dimensionalities.

4.3 Algorithms Compared

For comparison of the proposed algorithm we have considered 4 other algorithms which are: DynDE [16], dopt-aiNET [21] CPSO [22] and DASA [23]. For each of the algorithms taken for comparison, best parametric setup is employed as reported in paper.

Algorithm 3. MLDES Algorithm

begin

- 1 Initialize a population of Np individuals by means of SP subpopulations randomly. Hence number of individual reside in a subpopulation is, $n = (Np / SP)$.
- 2 Evaluate the population members and find the best member as $gbest$.
- 3 Insert a *test solution* in the search space and calculate its fitness value.
- 4 Set *threshold* as given in III-B.
- 5 $gen \leftarrow 1$
- 6 **while** the termination criterion is not satisfied
- 7 **for** $k=1:SP$
- 8 **for** $i=1:n$
- 9 Generate the mutant vector according to equation (9).
- 10 Crossover operation is performed according to (7).
- 11 Then fitness proportionate selection is done as given in (8) considering minimization problem. For a maximization problem the relational operator will be flipped.
- 12 **end for**
- 13 Find the best member in the subpopulation.
- 14 **end for**
- 15 Check the best individuals' pair wise distance of the subpopulations. If any falls below *threshold*, reinitialize the corresponding subpopulation which contains comparatively less fit subpopulation-best.
- 16 Evaluate *test solution*.
- 17 **If** $f(\text{test solution})_i \neq f(\text{test solution})_{i-1}$ **or**

4.4 Parametric Setup

For our algorithm we have used population size Np equal to 100. The crossover probability is set to 0.9. Total number of subpopulations is always kept at 10. The operating platform used for simulation has the following configurations:

- Programming Language: MATLAB R2010a.
- Operating System: Windows 7 Home Premium.
- Processor: Intel (R) Core™ i5 CPU 760@ 2.80 GHz
- Installed Memory: 2.0 GB.
- System Type: 32-bit Operating System.

For the MPB problem, we have considered 20 environmental changes in a single run and 25 such runs were conducted. For GDBG, for every test instance, 60 changes were considered in each run and 25 such runs were conducted. The results given in this paper are the means and standard deviation values of 25 independent runs.

Table 1. Default settings for the MPB Problem

| Parameter | Value |
|-----------------------------------|--------------|
| Number of Peaks | 10 |
| Change Frequency (E) | 5000 |
| Height severity | 7.0 |
| Width severity | 1.0 |
| Shift length s | 1.0 |
| Number of Dimensions D | 5 |
| Correlation coefficient λ | 0 |
| Search Range S | [0, 100] |
| H | [30.0, 70.0] |
| W | [1, 12] |
| Initial Height I | 50.0 |

5 Experimental Results

In this section we have provided the comparison results of MLDES with other peer algorithms in two set of benchmarks considered.

5.1 Experimental Study on MPB Problems

The comparison has been undertaken on various configurations of MPB problems. First, the result will be provided for different number of peaks and also with different dimensionality. The dimensionality is set at $D \in \{5, 10\}$ and the number of peaks is set, $p \in \{1, 10, 100\}$. The results are given in Table 2. Rest of the parameters is kept fixed at default values as mentioned in Table 1. In all the cases our algorithm managed first rank. Next, the shift severity will be varied keeping all other parameters at default values. Results of different algorithms are given in Table 3. The effect of varying change frequency is shown in Table 4. Here the change frequency E is taken from $\{3000, 5000, 10000\}$. In all the three cases the proposed algorithm has outperformed other four algorithms.

However, an experiment has also been conducted to study the comparative performance of different algorithms for different values of correlation coefficient. Other parameters of the MPB problem are kept fixed at default values. The means and standard deviations of the offline error values for all the algorithms are given in Table 5. As we can see that in all the cases, our algorithm has outperformed other four algorithms.

5.2 Experimental Study on GDBG System

GDBG system is more challenging due to the rotation of peaks along with the change of the location, height and width of the peaks. There are total 49 change instances. To measure the performance of the considered algorithms we have taken the metric: offline error and standard deviation. Offline error can be measured by:

$$e_{off} = \frac{1}{M \times N} \sum_{m=1}^M \sum_{n=1}^N e_{m,n}^{last} \quad (11)$$

$$SD = \sqrt{\sum_{m=1}^M \sum_{n=1}^N (e_{m,n}^{last} - e_{off}) / (M \times N - 1)} \tag{12}$$

where M and N are the total number of runs and total number of environment changes at each run respectively. $e_{m,n}^{last}$ is the error at the last generation.

Table 2. OFFLINE ERROR AND STANDARD DEVIATION FOR VARYING VALUES OF D AND P

| D, p | DASA | DynDE | dopt-aiNET | CPSO | MLDE-S |
|--------|----------------|----------------|----------------|----------------|------------------------------|
| 5,1 | 4.89 (0.07) | 5.05 (0.08) | 4.88 (0.07) | 1.54 (0.07) | 1.11 (0.07) |
| 5,10 | 1.70 (0.06) | 1.76 (0.06) | 1.75 (0.05) | 1.80 (0.06) | 1.35 (0.07) |
| 5,100 | 1.69 (0.05) | 2.55 (0.08) | 1.76 (0.06) | 1.92 (0.06) | 1.65 (0.08) |
| 10,1 | 5.12 (0.07) | 5.32 (0.08) | 5.02 (0.07) | 2.07 (0.06) | 1.88 (0.08) |
| 10,10 | 2.31 (0.07) | 3.19 (0.09) | 2.35 (0.06) | 2.39 (0.06) | 2.01 (0.09) |
| 10,100 | 2.47 (0.08) | 3.38 (0.09) | 2.59 (0.07) | 2.51 (0.07) | 2.25 (0.08) |

Table 3. OFFLINE ERROR AND STANDARD DEVIATION FOR VARYING S

| S | DASA | DynDE | dopt-aiNET | CPSO | MLDE-S |
|-----|----------------|----------------|----------------|----------------|------------------------------|
| 0 | 0.98 (0.07) | 1.15 (0.06) | 1.00 (0.06) | 0.97 (0.06) | 0.79 (0.06) |
| 1 | 1.70 (0.06) | 1.76 (0.06) | 1.75 (0.05) | 1.80 (0.06) | 1.35 (0.07) |
| 2 | 2.17 (0.07) | 2.27 (0.06) | 2.15 (0.06) | 2.23 (0.06) | 1.91 (0.08) |
| 3 | 2.75 (0.07) | 2.70 (0.06) | 2.76 (0.08) | 2.83 (0.08) | 2.34 (0.07) |
| 4 | 2.96 (0.07) | 3.29 (0.07) | 2.85 (0.08) | 3.09 (0.09) | 2.53 (0.08) |
| 5 | 3.00 (0.08) | 4.32 (0.08) | 2.90 (0.08) | 3.17 (0.09) | 2.68 (0.11) |
| 6 | 3.30 (0.08) | 4.55 (0.09) | 3.37 (0.08) | 3.45 (0.09) | 3.07 (0.10) |

Table 4. Offline Error and Standard Deviation of Algorithms for Varying Change Frequency (E)

| <i>E</i> | DASA | DynDE | dopt-aiNET | CPSO | MLDE-S |
|----------|----------------|----------------|----------------|----------------|------------------------------|
| 3000 | 2.42 (0.13) | 2.39 (0.17) | 2.40 (0.13) | 2.37 (0.13) | 2.09 (0.10) |
| 5000 | 1.70 (0.06) | 1.76 (0.06) | 1.75 (0.05) | 1.80 (0.06) | 1.35 (0.07) |
| 10000 | 0.97 (0.06) | 1.33 (0.04) | 0.98 (0.05) | 0.99 (0.05) | 0.88 (0.05) |

The mean and standard deviations of offline errors of the 49 instances is given in Table 6. Our proposed algorithm has outperformed the other four in almost all the cases except at T3 of F1 with number of peaks 10, F4-T4, F5-T4 and F6-T2. In the rest 46 cases our algorithm has managed to get first position.

6 Working Mechanism of MLDES

In this section the working mechanism MLDES will be investigated on the MPB problem. In Fig. 1, offline error is plotted against number of function evaluations. As we can see that the curve is decreasing in nature in a specific environment. Whenever an environmental change occurs, it jumps to a larger value because a part of the population is reinitialized as mentioned in the algorithm description.

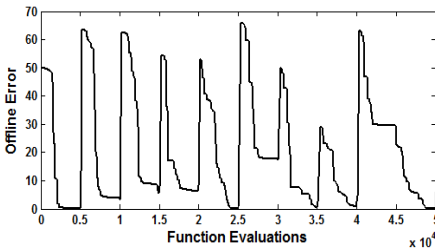


Fig. 1. Variation of Offline Error with number

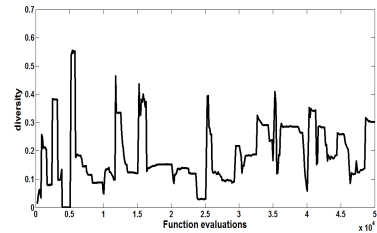


Fig. 2. Variation of population diversity w.r.t. function evaluations

Another important prospect of any DOP optimizer is to maintain the population diversity. The ‘distance-to-average point’ measurement [21] for the diversity of a particular population at instant *t* can be given as:

$$div(P(t)) = \frac{1}{N_p \times L} \sum_{i=1}^{N_p} \sqrt{\sum_{j=1}^D (x_{ij} - \bar{x}_j)^2} \tag{13}$$

Table 5. AVERAGE AND STANDARD DEVIATION OF ERRORS OBTAINED BY ALL THE ALGORITHMS IN GDBG BENCHMARK

| Test Functions | Algorithm | Error | T1 | T2 | T3 | T4 | T5 | T6 | T7 |
|----------------------------|------------|---------------|--------------------------|---------------------------|---------------------------|-------------------------|--------------------------|---------------------------|---------------------------|
| F1 (Number of peaks 10) | DASA | Average (Std) | 0.1854 (1.2501) | 4.1802 (9.0716) | 6.3706 (10.7128) | 0.4873 (1.9504) | 2.5485 (4.8002) | 2.3442 (8.6685) | 4.8452 (8.9662) |
| | DynDE | Average (Std) | 0.0732 (2.9567) | 2.5567 (8.4313) | 5.4245 (9.2485) | 0.1263 (0.9426) | 1.5651 (4.6461) | 1.3115 (6.2511) | 4.1137 (8.5249) |
| | dopt-aiNET | Average (Std) | 0.1353 (1.0061) | 5.8667 (10.2772) | 4.2545 (8.1828) | 5.3563 (8.9414) | 4.4356 (5.5545) | 9.9407 (15.8214) | 4.2110 (8.6873) |
| | CPSO | Average (Std) | 0.0351 (0.4262) | 2.7185 (6.5230) | 4.1315 (8.9947) | 0.0944 (0.7855) | 1.8698 (4.4910) | 1.1569 (4.8054) | 4.5401 (9.1194) |
| | MLDES | Average (Std) | 0.0006 (0.0005) | 1.0113 (4.4738) | 12.8764 (5.2314) | 0.0194 (0.0123) | 1.1381 (0.6126) | 1.0069 (0.4785) | 3.9557 (4.1185) |
| F1 (Number Of Peaks 50) | DASA | Average (Std) | 0.4425 (1.3911) | 4.8661 (7.0052) | 8.4247 (9.5682) | 0.5853 (1.0901) | 1.1832 (2.1818) | 2.0728 (5.9719) | 7.8412 (9.0558) |
| | DynDE | Average (Std) | 0.3286 (1.5224) | 4.6547 (6.3453) | 4.6441 (9.3523) | 0.1412 (0.5914) | 1.0162 (2.6489) | 0.9859 (4.8631) | 6.2513 (9.0651) |
| | dopt-aiNET | Average (Std) | 0.3644 (0.9725) | 4.7485 (6.7580) | 5.2531 (6.6830) | 2.6565 (5.9773) | 2.8641 (4.1579) | 6.8330 (11.8790) | 4.8172 (6.4528) |
| | CPSO | Average (Std) | 0.2624 (0.9362) | 3.2792 (5.3034) | 6.3198 (7.4420) | 0.1255 (0.3859) | 0.8481 (1.7790) | 1.4821 (4.3932) | 6.6467 (7.9411) |
| | MLDES | Average (Std) | 0.0375 (0.0045) | 2.3342 (4.3863) | 4.8374 (6.0813) | 0.0878 (0.1432) | 0.6612 (1.4021) | 0.0594 (2.8765) | 4.1852 (6.8432) |
| F2 | DASA | Average (Std) | 3.3017 (8.7885) | 25.6105 (83.2124) | 18.9904 (67.8204) | 1.4512 (3.8311) | 49.6022 (112.4132) | 2.1182 (5.2912) | 3.8752 (8.1285) |
| | DynDE | Average (Std) | 1.3627 (5.0315) | 13.0179 (48.2532) | 11.9214 (45.7054) | 0.7842 (2.2248) | 20.7842 (64.5341) | 2.1845 (3.9643) | 2.4235 (7.1031) |
| | dopt-aiNET | Average (Std) | 0.0984 (0.0291) | 8.1209 (14.3832) | 17.9979 (62.2259) | 1.0652 (2.8269) | 101.384 (134.5180) | 6.5192 (13.8172) | 3.7385 (7.9542) |
| | CPSO | Average (Std) | 1.2475 (4.1780) | 10.1055 (35.0601) | 10.2725 (33.4527) | 0.5664 (2.1371) | 25.1424 (64.2500) | 1.9871 (5.2175) | 3.6510 (6.9274) |
| | MLDES | Average (Std) | 0.0314 (0.1745) | 7.3337 (9.4355) | 7.1561 (9.4362) | 0.1418 (1.8754) | 19.7156 (50.5215) | 0.9154 (3.0713) | 2.1143 (9.2347) |
| F3 | DASA | Average (Std) | 15.7025 (67.1131) | 824.389 (204.0035) | 688.358 (298.0124) | 435.488 (441.2120) | 697.210 (315.4223) | 626.1120 (460.6211) | 433.252 (380.2234) |
| | DynDE | Average (Std) | 21.2512 (73.6549) | 792.457 (255.6163) | 635.614 (342.7753) | 341.701 (419.8116) | 749.265 (280.9181) | 519.550 (438.2467) | 415.324 (390.3450) |
| | dopt-aiNET | Average (Std) | 810.830 (66.1085) | 1078.70 (64.1245) | 1073.41 (64.9950) | 1031.54 (274.7490) | 1023.90 (57.8713) | 1186.94 (292.2960) | 1061.33 (110.0980) |
| | CPSO | Average (Std) | 137.527 (221.0011) | 855.139 (161.0024) | 765.966 (235.8834) | 430.620 (432.2391) | 859.704 (121.5581) | 753.039 (361.7855) | 653.703 (334.4892) |
| | MLDES | Average (Std) | 11.3834 (39.1034) | 547.153 (234.0025) | 596.344 (304.3887) | 77.667 (112.613) | 533.714 (48.8842) | 328.114 (244.8822) | 197.445 (264.7456) |
| F4 | DASA | Average (Std) | 5.6001 (26.5331) | 65.6105 (160.0281) | 53.6158 (140.0155) | 3.8512 (4.2258) | 118.212 (178.2506) | 2.9812 (7.5942) | 27.4432 (90.2513) |
| | DynDE | Average (Std) | 1.8616 (5.7531) | 39.5923 (98.6312) | 23.4921 (94.5314) | 0.9691 (3.1723) | 44.6713 (121.7162) | 1.5624 (6.2149) | 6.5213 (26.5951) |
| | dopt-aiNET | Average (Std) | 1.4227 (4.5459) | 122.440 (201.627) | 98.6688 (196.6950) | 4.2632 (9.7255) | 304.566 (203.2430) | 12.6491 (55.8367) | 52.9010 (130.593) |
| | CPSO | Average (Std) | 2.6771 (7.0552) | 37.1512 (99.4352) | 36.6711 (97.1805) | 0.7926 (2.775) | 67.1702 (130.3059) | 4.8814 (15.3965) | 12.7924 (19.2105) |
| | MLDES | Average (Std) | 1.3986 (5.0023) | 10.9394 (32.8612) | 2.0020 (4.1124) | 1.0823 (3.8752) | 55.2980 (94.6889) | 1.5231 (5.5465) | 10.0413 (8.9944) |
| F5 | DASA | Average (Std) | 0.9551 (3.4310) | 2.0199 (4.0504) | 0.9494 (3.3135) | 0.3928 (1.6184) | 2.3052 (6.3610) | 0.4671 (1.7346) | 1.1128 (3.7620) |
| | DynDE | Average (Std) | 2.9929 (6.8831) | 2.9481 (4.7179) | 2.9125 (5.3886) | 1.3796 (2.4199) | 8.4378 (12.1132) | 2.3049 (3.6182) | 0.5214 (0.7135) |
| | dopt-aiNET | Average (Std) | 40.8943 (221.212) | 34.4531 (119.896) | 34.9420 (115.025) | 120.637 (293.542) | 943.223 (633.318) | 480.305 (610.801) | 219.461 (427.817) |
| | CPSO | Average (Std) | 1.8559 (5.1812) | 2.8791 (6.7875) | 3.403 (6.4480) | 1.0954 (4.8651) | 7.9869 (13.8170) | 4.0535 (8.3719) | 6.5278 (22.8129) |
| | MLDES | Average (Std) | 0.1784 (2.5762) | 2.0393 (3.8845) | 0.4887 (2.1525) | 0.7321 (2.3456) | 1.0742 (3.3417) | 0.2145 (0.9843) | 0.2672 (0.6142) |
| F6 | DASA | Average (Std) | 8.8752 (13.319) | 37.128 (122.0147) | 26.7341 (98.4018) | 9.7442 (22.0541) | 37.9102 (118.0146) | 13.3481 (57.4802) | 17.7422 (36.7159) |
| | DynDE | Average (Std) | 6.0471 (11.0458) | 30.2205 (62.2093) | 19.3782 (67.3585) | 8.8731 (26.6683) | 43.3514 (136.9062) | 12.1784 (25.2617) | 15.3644 (21.0744) |
| | dopt-aiNET | Average (Std) | 20.4434 (79.3230) | 391.195 (395.4350) | 456.443 (405.0380) | 83.9698 (220.1770) | 845.862 (251.208) | 482.201 (434.421) | 372.740 (394.6628) |
| | CPSO | Average (Std) | 6.7254 (9.9747) | 31.5738 (63.5115) | 27.1358 (83.9873) | 9.2742 (24.2344) | 71.5704 (160.3211) | 23.6757 (51.5521) | 32.5842 (76.9105) |
| | MLDES | Average (Std) | 4.8973 (11.6342) | 38.2158 (58.7734) | 14.1145 (32.1304) | 6.1392 (10.9745) | 13.5174 (57.3466) | 6.6756 (8.0045) | 12.4414 (20.7219) |

where N_p is the total number of population in a D dimensional space. L is the length of longest diagonal in the search space and \bar{x}_j is the j^{th} dimensional component of the average vector. Variation of diversity with number of function evaluations is given in Fig. 2. As diversity diminishes, due to use of exclusion principle, diversity will be enhanced which is necessary for performing in dynamic environments. For using this speciation based memory archive after environment change detection, good quality solutions will be preserved and will be used in the ongoing generations. It enhances fine searching ability. In the default settings of MPB, the offline error with using speciation-based memory archive is 1.35 whereas without it is 1.94.

7 Conclusion

This paper lays down the foundation of a multipopulation model of DE for tackling dynamic optimization problems. The main crux of MLDEs is the synergism of local mutation strategy with the speciation memory archive. The local mutation scheme utilizes the concept of neighborhood and uses normal distribution to generate a vector that helps to balance exploration and exploitation in DE. As a diversity check, the exclusion principle is used to prevent wastage of FEs concerned with converged solutions. The advantage of the speciation based memory is the preservation of promising solutions of past environments which may serve as ideal search points in the changed environment. The proposed MLDEs technique is a novel approach to successfully tackle dynamism. The speciation memory archive can be extended to the field of dynamic niching. It can be used in conjunction with standalone niching algorithms for multiple peak detection in dynamic environments. The authors are presently working on it as ongoing research interest.

References

- [1] Storn, R., Price, K.V.: Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11(4), 341–359 (1997)
- [2] Jin, Y., Branke, J.: Evolutionary Optimization in Uncertain Environments- A Survey. *IEEE Transactions on Evol. Comput.* 9(3), 303–317 (2005)
- [3] Trojanowski, K., Michalewicz, Z.: Evolutionary Optimization in Non-Stationary Environments. *Journal of Computer Science and Technology* 1(2), 93–124 (2000)
- [4] Lampinen, J., Zelinka, I.: On stagnation of the differential evolution algorithm. In: Ošmera, P. (ed.) *Proc. of MENDEL 2000, 6th International Mendel Conference on Soft Computing*, Brno, Czech Republic, June 7-9, pp. 76–83 (2000)
- [5] Fogel, L.J., Owens, A.J., Walsh, M.J.: *Artificial Intelligence Through Simulated Evolution*. Wiley, New York (1966)
- [6] Branke, J.: *Evolutionary Optimization in Dynamic Environments*. Kluwer, Norwell (2001)
- [7] Yang, S., Ong, Y.S., Jin, Y. (eds.): *Evolutionary Computation in Dynamic and Uncertain Environments*. Springer, Berlin (2007)

- [8] Grefenstette, J.J.: Genetic algorithms for changing environments. In: Proc. 2nd Int. Conf. Parallel Problem Solving from Nature, pp. 137–144 (1992)
- [9] Yang, S.: Memory based immigrants for genetic algorithms in dynamic environments. In: Proc. 2005 Genetic and Evol. Comput. Conf., vol. 2, pp. 1115–1122 (2005)
- [10] Yang, S.: Associative memory scheme for genetic algorithms in dynamic environments. In: Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H. (eds.) *EvoWorkshops 2006*. LNCS, vol. 3907, pp. 788–799. Springer, Heidelberg (2006)
- [11] Blackwell, T.M., Bentley, P.J.: Dynamic search with charged swarms. In: Proc. 2002 Genetic Evol. Comput. Conf., pp. 19–26 (2002)
- [12] Biswas, S., Bose, D., Kundu, S.: A Clustering Particle Based Artificial Bee Colony Algorithm for Dynamic Environment. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Nanda, P.K. (eds.) *SEMCCO 2012*. LNCS, vol. 7677, pp. 151–159. Springer, Heidelberg (2012)
- [13] Bose, D., Biswas, S., Kundu, S., Das, S.: A Strategy Pool Adaptive Artificial Bee Colony Algorithm for Dynamic Environment through Multi-population Approach. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Nanda, P.K. (eds.) *SEMCCO 2012*. LNCS, vol. 7677, pp. 611–619. Springer, Heidelberg (2012)
- [14] Biswas, S., Kundu, S., Das, S., Vasilakos, A.V.: Information sharing in bee colony for detecting multiple niches in non-stationary environments. In: Blum, C. (ed.) *Proceeding of the Fifteenth Annual Conference Companion on Genetic and Evolutionary Computation Conference Companion (GECCO 2013 Companion)*, pp. 1–2. ACM, New York (2013), <http://doi.acm.org/10.1145/2464576.2464588>
- [15] Kundu, S., Biswas, S., Das, S., Suganthan, P.N.: “Crowding-based local differential evolution with speciation-based memory archive for dynamic multimodal optimization. In: Blum, C. (ed.) *Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference (GECCO 2013)*, pp. 33–40. ACM, New York (2013), <http://doi.acm.org/10.1145/2463372.2463392>
- [16] Mendes, R., Mohais, A.S.: DynDE: a differential evolution for dynamic optimization problems. In: Proc. of IEEE Congress on Evolutionary Computation, vol. 2, pp. 2808–2815 (2005)
- [17] Angira, R., Santosh, A.: Optimization of dynamic systems: A trigonometric differential evolution approach. *Computers & Chemical Engineering* 31(9), 1055–1063 (2007)
- [18] Lung, R.I., Dumitrescu, D.: A collaborative model for tracking optima in dynamic environments. In: Proc. 2007 Congr. Evol. Comput., pp. 564–567 (2007)
- [19] Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: Proc. of IEEE Congress on Evolutionary Computation, vol. 3, pp. 1875–1882 (1999)
- [20] 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)
- [21] de França, F.O., Von Zuben, F.J.: A dynamic artificial immune algorithm to challenging benchmarking problems. In: Proc. 2009 Congr. Evol. Comput., pp. 423–430 (2009)
- [22] Li, C., Yang, S.: A clustering particle swarm optimizer for dynamic optimization. In: Proc. 2009 Congr. Evol. Comput., pp. 439–446 (2009)
- [23] Hu, J., Zeng, J.C., Tan, Y.: A diversity-guided particle swarm optimizer for dynamic environments. In: Li, K., Fei, M., Irwin, G.W., Ma, S. (eds.) *LSMS 2007*. LNCS, vol. 4688, pp. 239–247. Springer, Heidelberg (2007); vol. 9(3), pp. 303–317 (June 2005)

A Modified Differential Evolution for Symbol Detection in MIMO-OFDM System

Aritra Sen¹, Subhrajit Roy¹ and Swagatam Das²

¹ School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore 639798

² Electronics and Communications Sciences Unit,
Indian Statistical Institute, Kolkata 700 108, India
ARITRA001@e.ntu.edu.sg, subhrajit.roy@ntu.edu.sg,
swagatam.das@isical.ac.in

Abstract. It is essential to estimate the Channel and detect symbol in multiple-input and multiple-output (MIMO)-orthogonal frequency division multiplexing (OFDM) systems. Symbol detection by applying the maximum likelihood (ML) detector gives excellent performance but in systems with higher number of antennas and greater constellation size, the computational complexity of this algorithm becomes quite high. In this paper we apply a recently developed modified Differential Evolution (DE) algorithm with novel mutation, crossover as well as parameter adaptation strategies (MDE- p BX) for reducing the search space of the ML detector and the computational complexity of symbol detection in MIMO-OFDM systems. The performance of MDE- p BX have been compared with two classical symbol detectors namely ML and ZF and two famous evolutionary algorithm namely SaDE and CLPSO.

1 Introduction

Orthogonal frequency division multiplexing (OFDM)[1] is a standard multi-carrier modulation in high data rate wireless as well as wired communication systems. OFDM has the potential to increase spectral efficiency. This attribute has recently attracted much attention to OFDM so that data rate transmission can be increased considerably in modern communication systems. Recent communication systems like WLAN, HIPERMAN and 4G wireless cellular systems [2] have multiple-input, multiple-output (MIMO) technology incorporated in them. Combining OFDM with such MIMO technology has resulted in a significant capacity increase in such systems. For coherent demodulation of the signal, we need channel estimation and symbol detection at the receiver of these systems. A number of algorithms, such as the maximum likelihood (ML) and zero forcing (ZF) algorithms [3,4], have been proposed to detect symbols in OFDM. The implementation of the ZF algorithm is not complex and it is not computationally tedious but it fails to perform satisfactorily in fast-fading and time-varying environments. For these environments we therefore use the ML algorithm which performs excellently in these cases. The primary disadvantage of

the ML algorithm is its extremely high computational complexity. By computing the Euclidean distance between the received and actual symbols for all possible combinations of the transmitted symbols, an exhaustive search of the candidate symbol vector on each subcarrier is made. As the number of transmitter and receiver antennas increase in the constellation, the search space grows exponentially. So the algorithms computational complexity becomes intensive [5]. There is substantial literature which focuses on the area of reducing the complexity and obtaining an optimal solution from the ML algorithm, which detects symbols. Sphere decoders are used for ML detection of signals at the receiver end for multi fading channels. [6] has used orthogonal matrix triangularization (QR decomposition) with sort and Dijkstras algorithm for decreasing the computational complexity of these decoders. The multistage likelihood was presented in [7] to calculate the Euclidean distance of the candidate symbol. In [8], the sphere detector was proposed to have a polynomial computational reduction, but when the search space is large, it takes much more computational time. So research has focussed particularly in reducing the search space and decreasing the computational complexity. So for channel estimation and symbol detection heuristic approaches such as the genetic algorithm (GA) and particle swarm optimization (PSO) are implemented with the ML principle for their ability to reduce the computational burden by efficiently shrinking the search space. In pulse amplitude modulation-based communication systems (PAM) the GA [9] and PSO [10] strategies were used for channel estimation and data detection. A memetic differential evolution (DE) algorithm was proposed for minimum bit error rate detection in multiuser MIMO systems in [13]. DE [11] was also employed by [12] to reduce the computational complexity of symbol detectors in MIMO-OFDM system. In this paper, we employ a recently developed variant of DE called MDE- p BX [13], which introduced a new group-based mutation strategy, novel schemes for the adaptation of control parameters scale factor (F) and crossover rate (Cr), and also an exploitative crossover strategy (p -best crossover). The results of MDE- p BX has been compared to ML, ZF, SaDE [14] and CLPSO [15]. The results clearly indicate that MDE- p BX can obtain high quality results better than ZF, SaDE and CLPSO under various simulation strategies. Moreover, the results obtained by MDE- p BX are very close to the optimal results provided by ML algorithm, but MDE- p BX is computationally much less intensive than ML especially for larger systems.

2 Theory

2.1 MIMO-OFDM System Model

The simplified block diagram of the MIMO-OFDM system is shown in Figure 1. For this system, we consider the N_{tx} transmit, N_{rx} receive antennas, K subcarriers, n OFDM symbols. Considering the modulation type a vector of the information data is mapped onto complex symbols. The transmitted symbol vector is expressed as:

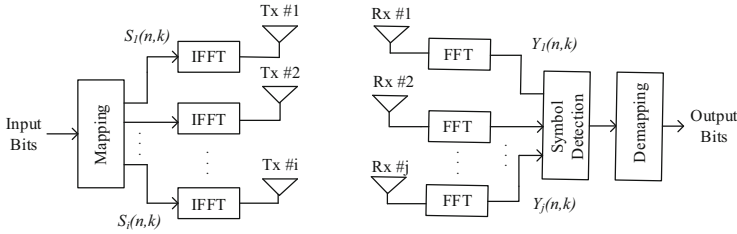


Fig. 1. Basic block diagram of MIMO-OFDM System

$$S[n, k] = [S_1(n, k), \dots, S_{N_{tx}}(n, k)]^T, \quad k = 0, \dots, K - 1, \quad (1)$$

where $S_i(n, k)$ is the symbol that is transmitted at the n^{th} symbol, k^{th} sub-carrier, and i^{th} antenna, and $[\cdot]^T$ is the transpose operation. By applying inverse fast Fourier transform (IFFT), symbol vectors are turned into the OFDM symbol:

$$s_n[m] = \frac{1}{\sqrt{KN_{tx}}} \sum_{k=0}^{K-1} S[n, k] e^{j2\pi m/k}, \quad m = 0, \dots, K - 1 \quad (2)$$

Then we add the cyclic prefix (CP) to avoid inter-symbol interference (ISI). The signal vectors are fed through the i^{th} transmitter antenna. After removing the CP from the received signal vector at the q^{th} receiver antenna, the fast Fourier transform (FFT) is taken as:

$$Y[n, k] = \frac{1}{\sqrt{K}} \sum_{m=0}^{K-1} y[m] e^{-j2\pi m/K}, \quad n = 0, \dots, K - 1 \quad (3)$$

Next, the received signal vector can be expressed as:

$$Y_q[n, k] = \sum_{i=1}^{N_{tx}} H_i[n, k] S_i[n, k] + W_q[n, k] \quad (4)$$

where $H_i[n, k]$ is the channel impulse response vector and $W_q[n, k]$ is the additive white Gaussian noise [14].

2.2 Symbol Detection in MIMO-OFDM

The estimations of the data symbols are obtained by maximizing the following metric:

$$S_* = \arg \max P(Y|S) \quad (5)$$

Furthermore, the ML algorithm detects the symbols by minimizing the squared Euclidian distance to target vector Y over the N_{tx} dimensional discrete search set:

$$S_* = \arg \min ||Y - HS||^2 \quad (6)$$

All possible $M^{N_{tx}}$ combinations of the transmitted symbols must be searched for the optimal solution of the ML detection which increases the computational complexity. Hence, we propose heuristic approaches in order to reduce the computational complexity of the symbol detection in the MIMO-OFDM systems.

3 Differential Evolution

DE [11,16] is a simple real-coded evolutionary algorithm. It works through a simple cycle of stages, which are detailed below.

3.1 Parameter Vector Initialization

DE searches for a global optimum point in a D -dimensional continuous hyper-space. It begins with a randomly initiated population of NP D dimensional real-valued parameter vectors. Each vector, also known as chromosome, forms a candidate solution to the multi-dimensional optimization problem. We shall denote subsequent generations in DE by $G = 0, 1, \dots, G_{max}$. Since the parameter vectors are likely to be changed over different generations, we may adopt the following notation for representing the i^{th} vector of the population at the current generation:

$$\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, \dots, x_{D,i,G}] \tag{7}$$

The initial population (at $G = 0$) should cover the entire search space as much as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum bounds: $\vec{X}_{min} = [x_{1,min}, x_{2,min}, \dots, x_{D,min}]$ and $\vec{X}_{max} = [x_{1,max}, x_{2,max}, \dots, x_{D,max}]$. Hence we may initialize the j^{th} component of the i^{th} vector as:

$$x_{j,i,0} = x_{j,min} + rand_{i,j}[0, 1] \cdot (x_{j,max} - x_{j,min}) \tag{8}$$

where $rand$ is a uniformly distributed number lying between 0 and 1 and is instantiated independently for each component of the i^{th} vector.

3.2 Mutation with Difference Vectors

After initialization, DE creates a donor vector V_{in} corresponding to each population member or target vector $X_{i,G}$ in the current generation through mutation. Three most frequently referred mutation strategies implemented in the public-domain DE codes available online at <http://www.icsi.berkeley.edu/storn/code.html> are listed below:

$$DE/rand/1 : \vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) \tag{9}$$

$$DE/best/1 : \vec{V}_{i,G} = \vec{X}_{r_{best}^i,G} + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \tag{10}$$

$$DE/target-to-best/1 : \vec{V}_{i,G} = \vec{X}_{i,G} + F \cdot (\vec{X}_{r_{best}^i,G} - \vec{X}_{i,G} + \vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \quad (11)$$

The indices r_1^i , r_2^i and r_3^i are mutually exclusive integers randomly chosen from the range $[1, NP]$, and all are different from the index i . These indices are randomly generated once for each donor vector. The scaling factor F is a positive control parameter for scaling the difference vectors. $\vec{X}_{r_{best}^i,G}$ is the best individual vector with the best fitness in the population at generation G .

3.3 Crossover

To enhance the potential diversity of the population, a crossover operation comes into play after generating the donor vector through mutation. The donor vector exchanges its components with the target vector $\vec{X}_{i,G}$ under this operation to form the trial vector $\vec{U}_{i,G} = [u_{1,i,G}, u_{2,i,G}, \dots, u_{D,i,G}]$. In this article we focus on the widely used binomial crossover that is performed on each of the D variables whenever a randomly generated number between 0 and 1 is less than or equal to the Cr value. The scheme may be outlined as:

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } rand_{i,j}[0, 1] \leq Cr \text{ or } j = j_{rand} \\ x_{j,i,G} & \text{otherwise} \end{cases} \quad (12)$$

where, as before, $rand_{i,j}[0, 1]$ is a uniformly distributed random number, which is called anew for each j^{th} component of the i^{th} parameter vector. $j_{rand} \in 1, 2, \dots, D$ is a randomly chosen index, which ensures that $\vec{U}_{i,G}$ obtains at least one component from $\vec{V}_{i,G}$.

3.4 Selection

The next step of the algorithm calls for selection to determine whether the target or the trial vector survives to the next generation, i.e., at $G = G+1$. The selection operation is described as:

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G} & \text{if } f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G}) \\ \vec{X}_{i,G} & \text{if } f(\vec{U}_{i,G}) > f(\vec{X}_{i,G}) \end{cases} \quad (13)$$

where $f(\vec{X})$ is the objective function to be minimized. Therefore, if the new trial vector gives an equal or lower value of the objective function, it replaces the corresponding target vector in the next generation; otherwise, the target is retained in the population.

4 The MDE- p BX Algorithm

We describe MDE- p BX and discuss the various features of the algorithm such as the mutation scheme called DE/current-to-gr_best/1, a p -best crossover scheme in this section. We also coin the rules for adapting the control parameters F and Cr in each iteration.

4.1 DE/current-to-gr_best/1

DE/current-to-best/1 is one of the widely used mutation schemes in DE. The useful information of the best solution (with highest objective function value for maximization problems) is incorporated in this algorithm. This results in fast convergence by guiding the evolutionary search towards a specific point in the search space. The algorithm may converge to a locally optimal point in the search space due to its exploitative behavior and lose its global exploration capabilities. To avoid this predicament, in this paper we propose a less greedy and more explorative variant of the DE/current-to-best/1 mutation strategy termed as DE/current-to-gr_best/1. This utilizes the best vector of a dynamic group of $q\%$ of the randomly selected population members for each target vector. Now the population moves towards different points and explores the landscape much better without getting attracted towards a specific point in the search space. The new scheme may be formulated as:

$$\vec{V}_{i,G} = \vec{X}_{i,G} + F \cdot (\vec{X}_{gr_best,G} - \vec{X}_{i,G} + \vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \tag{14}$$

where $\vec{X}_{gr_best,G}$ is the best solution of $q\%$ members randomly selected from the present population whereas $\vec{X}_{r_1^i,G}$ and $\vec{X}_{r_2^i,G}$ are two randomly selected distinct population vectors. Using the above technique, the target solutions are prevented from being attracted towards the same best position found so far by the entire population. This helps in avoiding premature convergence at local optima.

4.2 The p -best Crossover

The crossover operation in MDE- p BX is named p -best crossover where for each donor vector, a vector is randomly chosen from the p top-ranking individuals (in accordance with their objective function values) in the current population and then normal binomial crossover is carried out between the donor vector and the randomly selected p best vector to produce the trial vector of same index. Fast convergence is ensured by means of this innovative crossover scheme, where the information contained in the top ranking individuals of the population is incorporated into the trial vector. The parameter p is reduction takes place in a linear fashion in the following way:

$$p = ceil[N_p/2(1 - \frac{G - 1}{G_{max}})] \tag{15}$$

where N_p is the population size, G is the current generation number, G_{max} is the maximum number of generations ($G = [1, 2, \dots, G_{max}]$) and $ceil(y)$ is the ‘ceiling function returning the lowest integer greater than its argument y . p is reduced by a routine which favours exploration at the initial stages of the search and exploitation during the later stages. This is done by gradually reducing the elitist portion of the population, with a randomly selected member from where the component mixing of the donor vector is allowed for generation of the trial vector.

4.3 Parameter Adaptation

The parameter adaptation schemes in MDE-*p*BX are guided by the knowledge of the successful values of F and Cr that were able to generate better offspring (trial vectors) in the last generation.

Scale Factor Adaptation. At every generation, the scale factor F_i of each individual target vector is independently generated as $F_i = Cauchy(F_m, 0.1)$ where $Cauchy(F_m, 0.1)$ is a random number sampled from a Cauchy distribution with location parameter F_m and scale parameter 0.1. The value of F_i is regenerated if $F_i \leq 0$ or $F_i > 1$. Let us denote $F_{success}$ as the set of the successful scale factors, so far, of the current generation generating better trial vectors that are likely to advance to the next generation. Moreover let us say $mean_A(F_{G-1})$ is the simple arithmetic mean of all scale factors associated with population members in generation $G-1$. Location parameter F_m of the Cauchy distribution is initialized to be 0.5 and then updated at the end of each generation in the following manner:

$$F_m = w_F F_m + (1 - w_F) mean_{Pow}(F_{success}) \quad (16)$$

The weight factor though in the original MDE-*p*BX paper was varied randomly, but in our case we have set it as 0.8. $mean_{Pow}$ stands for power mean which is given by:

$$mean_{Pow}(F_{success}) = \sum_{x \in F_{success}} (x^n / |F_{success}|)^{1/n} \quad (17)$$

where $|F_{success}|$ is the cardinality of the set $F_{success}$ and n is taken as 1.5.

Crossover Probability Adaptation. At every generation the crossover probability Cr_i of each individual vector is independently generated as $Cr_i = Gaussian(Cr_m, 0.1)$, where $Gaussian(Cr_m, 0.1)$ is a random number sampled from a Gaussian distribution with mean Cr_m and standard deviation 0.1. Cr_i is truncated if it falls outside the interval $[0, 1]$. Denote $Cr_{success}$ as the set of all successful crossover probabilities Cr_i s at the current generation. The mean of the normal distribution Cr_m is initialized to be 0.6 and then updated at the end of each generation as:

$$Cr_m = w_{Cr} Cr_m + (1 - w_{Cr}) mean_{Pow}(Cr_{success}) \quad (18)$$

with the weight w_{Cr} being kept constant at 0.9. The power mean is calculated as:

$$mean_{Pow}(Cr_{success}) = \sum_{x \in Cr_{success}} (x^n / |Cr_{success}|)^{1/n} \quad (19)$$

where $|Cr_{success}|$ is the cardinality of the set $Cr_{success}$ and n is taken as 1.5.

5 Experiments and Results

For our experimentation we have considered three OFDM systems with 2×4 , 4×4 and 8×8 transmitter and receiver antennas. We have compared the performance of MDE- p BX with two classical symbol detectors namely ML and ZF and two other renowned heuristic algorithms SaDE and CLPSO. The simulation parameters of the MIMO-OFDM system are presented in Table 1. The population size was kept as 25 for all the participating evolutionary algorithms. The parameter q in the mutation scheme DE/current-to-gr_best/1 of MDE- p BX is kept as 1/4th of the population size. The reason for setting such a value for the group size q is that if q is on par with population size, the probability that the best of randomly chosen $q\%$ vectors is similar to the globally best vector of the entire population will be high and the proposed mutation scheme DE/current-to-gr_best/1 basically becomes identical to the DE/current-to-best/1 scheme. This drives most of the vectors towards a specific point in the search space resulting in premature convergence. The parameter p in p -best crossover is linearly decreased with generations. For the contestant heuristic algorithms, we follow the parameter settings in the original paper of SaDE and CLPSO. The convergence analysis of MDE- p BX, SaDE and CLPSO are done in Fig. 2 where the number of iterations needed by these algorithms for converging to the optimal solution (obtained by ML) are shown for the three OFDM systems. were found out by averaging over 25 runs. As the same algorithm converges to the optimal solution at a different iteration in each simulation so the data presented in Fig. 2 are averaged over 25 runs. It is evident from Fig. 2 that MDE- p BX requires less number of iterations than SaDE and CLPSO for converging to the optimal solution for all the three cases.

Table 1. MIMO-OFDM simulation parameters

| Parameters | Values |
|-----------------------|-----------------|
| Number of subcarriers | 128 |
| Cyclic prefix size | FFT/4=32 |
| Modulation type | 8PSK |
| Channel type | Rayleigh fading |

The BER performance of the symbol detectors for the 2×4 system are shown in Fig. 3. The figure depicts that MDE- p BX outperforms SaDE, CLPSO and ZF and is very close to the optimal solution i.e. the BER provided by ML. For example, at 25 dB SNR the Bit Error Rates for ML, MDE- p BX, SaDE, CLPSO and ZF are 2.15×10^{-4} , 4×10^{-4} , 6×10^{-4} , 10^{-3} and 0.05 respectively.

Fig. 4 and Fig. 5 shows the BER performance of the symbol detectors for 4×4 and 8×8 systems respectively. These two figures show the effect of the increasing number of receiver and transmitter antennas on the detection performance. For instance, for the 4×4 system at 15 dB SNR the BERs for ML, MDE- p BX, SaDE, CLPSO and ZF are 5.95×10^{-3} , 8×10^{-3} , 9×10^{-3} , 1.3×10^{-2} and

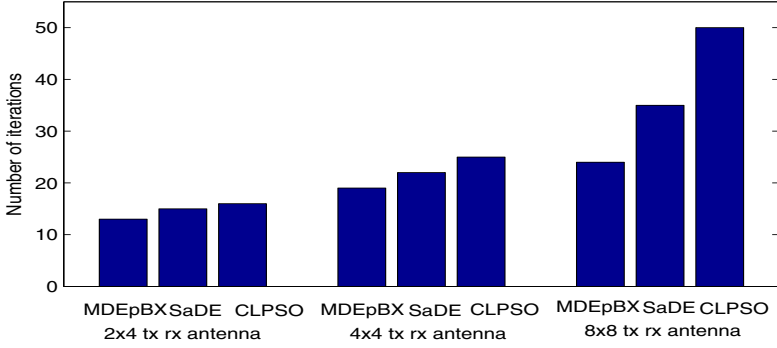


Fig. 2. Convergence analysis of MDE-pBX, SaDE and CLPSO

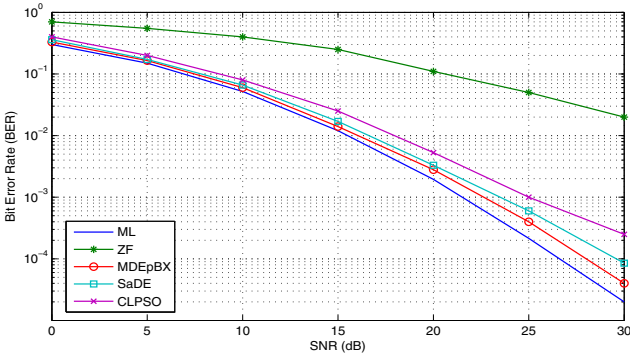


Fig. 3. The BER versus the SNR of the detectors for 2x4 system

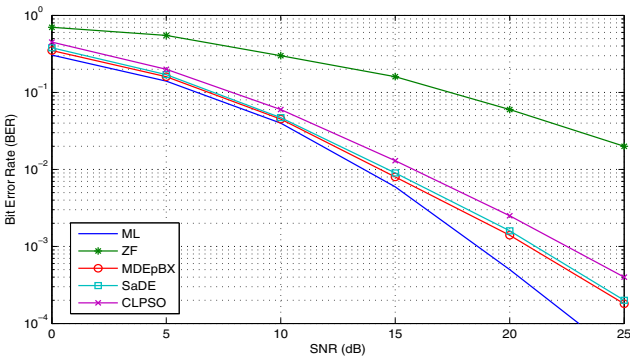


Fig. 4. The BER versus the SNR of the detectors for 4x4 system

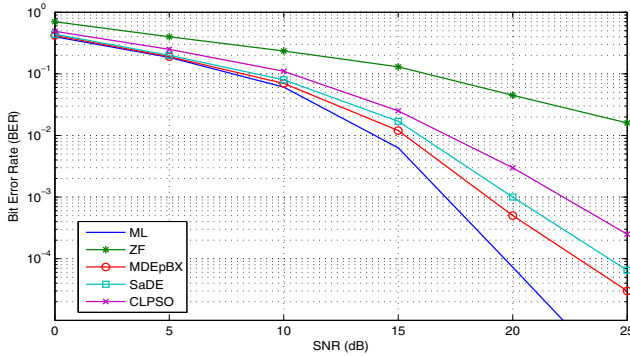


Fig. 5. The BER versus the SNR of the detectors for 8x8 system

0.16 respectively. Again, for the 8×8 system at 15 dB SNR the BERs for ML, MDE_pBX, SaDE, CLPSO and ZF are 6.3×10^{-3} , 1.2×10^{-2} , 1.7×10^{-2} , 2.5×10^{-2} and 0.13 respectively.

6 Computational Complexity

To show that the heuristic algorithms especially MDE_pBX are computationally advantageous than the classical symbol detectors we have done computational complexity analysis. The computational complexity of the symbol detectors have been represented in terms of N_{rx} (number of receiver antenna), N_{tx} (number of transmitter antenna), N_p (Population size), N_{itr} (Number of iterations) and M (Constellation size). The number of operations required for ZF and ML algorithms are $4N_{tx}^3 + 2N_{tx}^2N_{rx}$ and $N_{rx}(N_{tx} + 1)M^{N_{tx}}$ respectively. On the other hand the heuristic algorithms require $N_p(N_{tx}N_{rx} + \mu)N_{itr}$ operations, where μ is the number of population updating parameters. μ depends on the algorithm and is almost same for the heuristic approaches. But, the computational complexity does not directly depend on μ . It directly depends on the number of iterations that provide convergence in the algorithms. As for all the three systems, MDE_pBX requires the least number of iterations among the heuristic approaches so it has the lowest computational complexity. Moreover, it can be also seen from the above analysis that the computational complexity of the ML algorithm is very high when the number of transmitter antenna, receiver antenna and the constellation size increases. For this reason, the ML algorithm is not a practical solution for symbol detection in MIMO-OFDM systems that have large antenna and constellation sizes. However, the proposed MDE_pBX based detector has significantly less computational complexity than the other algorithms.

7 Conclusion

In this article, we utilized a DE variant, MDE- p BX, for reducing the search space of the ML detector and the computational complexity of symbol detection in MIMO-OFDM systems. We have compared the performance of MDE- p BX to two classical (ML and ZF) and two evolutionary algorithms (SaDE and CLPSO). Our results show that ML has the best performance followed by MDE- p BX, SaDE, CLPSO and ZF. However, though ML produces best results it is computationally very inefficient. Moreover, among the evolutionary algorithms MDE- p BX takes least number of iterations to reach the optimal solution i.e. the solution produced by ML. Thus, we can conclude that MDE- p BX can be a very good trade off between results and computational complexity.

References

1. Cimini, L.J.: Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing. *IEEE Transactions on Communications* 33(7), 665–675 (1985)
2. van Zelst, V., Schenk, T.C.W.: Implementation of a mimo ofdm-based wireless lan system. *IEEE Transactions on Signal Processing* 52(2), 483–494 (2004)
3. Kiessling, M., Speidel, J.: Analytical performance of mimo zero-forcing receivers in correlated rayleigh fading environments. In: 4th IEEE Workshop on Signal Processing Advances in Wireless Communications, SPAWC 2003, pp. 383–387 (2003)
4. Spencer, Q., Swindlehurst, A., Haardt, M.: Zero-forcing methods for downlink spatial multiplexing in multiuser mimo channels. *IEEE Transactions on Signal Processing* 52(2), 461–471 (2004)
5. Jiang, M., Akhtman, J., Guo, F.C., Hanzo, L.: Iterative joint channel estimation and symbol detection for multi-user mimo ofdm. In: *IEEE VTC 2006*, May 7–10, vol. 1, pp. 221–225 (Spring 2006)
6. Fukatani, T., Matsumoto, R., Uyematsu, T.: Two methods for decreasing the computational complexity of the mimo ml decoder. *CoRR cs.IT/0410040* (2004)
7. Xie, Y., Li, Q., Georgiades, C.: On some near optimal low complexity detectors for mimo fading channels. *IEEE Transactions on Wireless Communications* 6(4), 1182–1186 (2007)
8. Hassibi, B., Vikalo, H.: On the sphere-decoding algorithm i. expected complexity. *IEEE Transactions on Signal Processing* 53(8), 2806–2818 (2005)
9. Chen, S., Wu, Y.: Maximum likelihood joint channel and data estimation using genetic algorithms. *IEEE Transactions on Signal Processing* 46(5), 1469–1473 (1998)
10. Zubair, M., Choudhry, M.A.S., Malik, A.N., Qureshi, I.M.: Joint channel and data estimation using particle swarm optimization. *IEICE Transactions* 91-B(9), 3033–3036 (2008)
11. Storn, R., Price, K.: Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11(4), 341–359 (1997)
12. Seyman, M.N., Taspinar, N.: Symbol detection using the differential evolution algorithm in mimo-ofdm systems. *Turkish Journal of Electrical Engineering & Computer Sciences* (2013)

13. Islam, S., Das, S., Ghosh, S., Roy, S., Suganthan, P.: An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 42(2), 482–500 (2012)
14. Qin, A.K., Huang, V.L., Suganthan, P.: Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation* 13(2), 398–417 (2009)
15. Liang, J.J., Qin, A.K., Suganthan, P., Baskar, S.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation* 10(3), 281–295 (2006)
16. Yang, X.S., Deb, S.: Two-stage eagle strategy with differential evolution. *International Journal of Bio-inspired Computation* 4(1), 1–5 (2012)

Lèvy Flight Based Local Search in Differential Evolution

Harish Sharma¹, Shimpi Singh Jadon¹, Jagdish Chand Bansal²,
and K.V. Arya¹

¹ ABV-Indian Institute of Information Technology Gwalior, India

² South Asian University, New Delhi, India

{harish.sharma0107,shimpisingh2k6,jcbansal,kvarya}@gmail.com
<http://www.iiitm.ac.in>

Abstract. In order to solve non convex and complex optimization problems, Nature Inspired algorithms are being preferred in present scenario. Differential Evolution (DE) is relatively popular and simple population based probabilistic algorithm under the said category to find optimum value. The scale factor (F) and crossover probability (CR) are the two parameters which controls the performance of DE in its mutation and crossover processes by maintaining the balance between exploration and exploitation in search space. Literature suggests that due to large step sizes, DE is less capable of exploiting the existing solutions than the exploration of search space. Therefore unlike the deterministic methods, DE has inherent drawback of skipping the true optima. This paper incorporates the Levy Flight inspired local search strategy with DE named as Levy Flight DE (LFDE) which exploits the search space identified by best solution. To see the performance of LFDE, experiments are carried out on 15 benchmark problems of different complexities and results show that LFDE is a competitive DE variant and perform better than the basic DE and its recent variants namely Fitness based DE (FBDE) and Scale Factor Local Search DE (SFLSDE) in most of the test functions.

Keywords: Optimization, Nature Inspired Algorithms, Memetic Algorithm, Levy Flight based local search.

1 Introduction

Differential Evolution (DE) came in optimization environment in 1995 due to Storn and Price [19]. Differential Evolution is population based probabilistic and relatively a simple approach to find an optimum to the optimization problem. However, DE is an Evolutionary Algorithm (EA), but to some extent it significantly differs from EAs, e.g. DE first mutates the solution to generate the trail vector then it operates crossover operator to generate an offspring, while in EAs, crossover is applied first and then mutation. Also, in DE, to generate a trial vector, the distance and direction from current population like information is used.

DE is being tried to improve continuously. The recent modified DE versions with appropriate applications can be found in [2]. Experimental work in [20] indicates that DE outperforms the particle swarm optimization (PSO) [5] and genetic algorithm (GA) [4]. Almost in all fields of science and technology like mechanical engineering design [16], signal processing [3], chemical engineering [8] and specially in machine intelligence and pattern recognition [12], DE is been the preferred technique.

The variation and selection process are the two engines, which move the DE population's evolution towards the goal. However, without converging to local optima, DE is easily suspected to stop moving towards global optima [6]. These limitations demand to incorporate a local search approach in DE strategy to exploit the search region and hence to set the equilibrium between exploration and exploitation capabilities of DE. In this paper, Levy Flight random walk based local search strategy is proposed and assimilated with DE. The proposed strategy iteratively reduces the step size of the candidate solution movement in the search space within which the optima is known to exist. And hence can be used to get the global optima of a unimodal and/or multimodal functions. Further, the proposed strategy is distinguished on 15 test problems to the basic DE and its recent variants named, Fitness based DE (FBDE) [17] and Scale factor Local Search Differential Evolution (SFLSDE) [11].

Rest of the paper is organized as follows: Section 2 describes the Levy flight based local search strategy. Levy flight based local search in DE (LFDE) is proposed in section 3. In Section 4, experimental results are discussed. Finally, in section 5, paper is concluded.

2 Levy Flight Based Local Search

While adopting the local search techniques in general global search algorithms, all or some individuals of the generation try to identify better solutions in their small neighborhoods. So, these techniques are implemented in between the running iterations of population based search algorithms. The hybridized form of population based global search algorithms with local search techniques is called as memetic algorithms. In these algorithms, the global search ability tries to get the most promising regions in the search space, while the incorporated local search strategy examines closely the surroundings of some already found regions i.e., focuses rightly on exploitation. Recently, Sharma et al. [18] proposed a lévy flight local search (LFLS) strategy and incorporated the proposed strategy with ABC algorithm. In this paper, the LFLS strategy is hybridized with DE algorithm to enhance its exploitation capability.

The Levy Flight is a random walk in which the amount of movement is defined as step-length. The random step lengths are drawn from a certain probability distribution called Lévy distribution and defined as equation (1) [18]:

$$L(s) \sim |s|^{-1-\beta}, \text{ where } \beta (0 < \beta \leq 2) \text{ is an index and } s \text{ is the step length. (1)}$$

The paper utilizes Mantegna algorithm [21] to generate random step sizes using a symmetric Lévy stable distribution. Here the term 'symmetric' is mean

to say that the step size may be positive or negative. In Mantegna's algorithm, the step length s is calculated by

$$s = \frac{u}{|v|^{1/\beta}}, \quad (2)$$

where u and v are normally distributed numbers i.e., $u \sim N(0, \sigma_u^2)$, $v \sim N(0, \sigma_v^2)$ here,

$$\sigma_u = \left\{ \frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\beta \Gamma[(1 + \beta)/2] 2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad \sigma_v = 1. \quad (3)$$

This distribution (for s) obeys the expected lévy distribution for $|s| \geq |s_0|$, where s_0 is the smallest step length [21]. Here $\Gamma(\cdot)$ is the Gamma function and calculated as follows:

$$\Gamma(1 + \beta) = \int_0^\infty t^\beta e^{-t} dt. \quad (4)$$

In a special case when β is an integer, then we have $\Gamma(1 + \beta) = \beta!$.

In the proposed strategy, lévy distribution is used to generate the step sizes to exploit the search area and calculated as follows:

$$step_size(t) = 0.01 \times s(t) \times SLC, \quad (5)$$

here t is the iteration counter for local search strategy, $s(t)$ is calculated using lévy distribution as shown in equation (2), SLC is the social learning component of the global search algorithm and 0.01 is used to reduce the step size so that new solutions remains in search space.

The solution update equation based on the proposed local search strategy is:

$$x'_{ij}(t + 1) = x_{ij}(t) + step_size(t) \times U(0, 1), \quad (6)$$

here x_{ij} is the current position of i^{th} individual and $step_size(t) \times U(0, 1)$ is the actual amount of flight drawn from lévy distribution which is being added to move individual on next position.

The pseudo-code of the proposed *LFSL* is shown in Algorithm 1 [18]. In Algorithm 1, ϵ determines the termination of local search.

3 Levy Flight Based Local Search in DE (LFDE)

Population based optimization algorithms work well if they can maintain both the exploration and the exploitation as well. Finding out the different unknown segments of the the solution search space is the exploration, while, searching new potentials nearby already explored regions is the exploitation. Step size in DE algorithm, plays an important role because both exploration and exploitation can be done through it. The higher step size is responsible for exploration but can skip true solution, while, low step size results in exploitation and slow convergence. So the favorable variation in step sizes must be achieved in order

Algorithm 1. Lévy Flight Local Search:

```

Input optimization function  $Minf(x)$  and  $\beta$ ;
Select an individual  $x_i$  in the swarm which is going to be modified;
Initialize  $t = 1$  and  $\sigma_v = 1$ ;
Compute  $\sigma_u$  using equation (3);
while ( $t < \epsilon$ ) do
  Compute  $step\_size$  using equation (5);
  Generate a new solution  $x'_i$  using equation (6);
  Calculate  $f(x'_i)$ ;
  if  $f(x'_i) < f(x_i)$  then
     $x_i = x'_i$ ;
  end if
   $t = t + 1$ ;
end while

```

to improve the performance of the algorithm. Most of the algorithms in the optimization environment are lacking in this direction and so is the case with DE i.e., basic DE can not be suggested as an efficient algorithm [10, 6, 10].

Therefore, to achieve the same, Levy Flight based local search (*LFLS*) (described in section 2) has been applied in between the basic DE. In the proposed local search technique, only the best individual (say i^{th}) in current population is allowed to search better solution in its neighborhood. The pseudo-code of the proposed LFDE is shown in algorithm 2.

In Algorithm 2, ϵ is the termination criteria for the proposed local search. CR (same as crossover probability in binomial crossover) is a perturbation rate (a number between 0 and 1) which controls the amount of perturbation in the best solution, $U(0, 1)$ is a uniform distributed random number between 0 and 1, D is the dimension of the problem and x_k is a randomly selected solution within population. See section 4.2 for details of these parameter settings. The *LFLS* is applied in the basic DE to overcome the drawback of DE search process and named as Levy Flight based local search in Differential Evolution (*LFDE*). Pseudo-code of the *LFDE* is shown in Algorithm 3.

4 Experimental Results and Discussion

4.1 Test Problems Under Consideration

Table 1 shows the list of 15 different global optimization problems (f_1 to f_{15}) [1], which are taken under consideration, to figure out the performance of *LFDE*. All considered problems are of minimization type in nature and each has different degree of complexity and multimodality.

Algorithm 2. Lévy Flight based local Search in DE:

Input β , s and the best solution x_{best} of the current population;
 Initialize $t = 1$ and $\sigma_v = 1$;
 Compute σ_u using equation (3);
while ($t < \epsilon$) **do**
 Compute $step_size$ using equation (5);
 Generate a new solution x'_{best} as;
 for $j = 1$ to D **do**
 if $U(0, 1) < CR$ **then**
 $x'_{bestj} = x_{bestj} + 0.01 \times s \times (x_{bestj} - x_{kj}) \times U(0, 1)$;
 else
 $x'_{bestj} = x_{bestj}$;
 end if
 end for
 Calculate $f(x'_{best})$;
 if $f(x'_{best}) < f(x_{best})$ **then**
 $x_{best} = x'_{best}$;
 end if
 $t = t + 1$;
end while

Algorithm 3. Lévy Flight based local search in Differential Evolution (*LFDE*):

Initialize the control parameters F and CR ;
 Create and initialize the population $P(0)$ of NP individuals;
while Termination criteria **do**
 for each individual $x_i(G) \in P(G)$ **do**
 Evaluate the fitness $f(x_i(G))$;
 Generate the trial vector $u_i(G)$ by applying the mutation operator;
 Generate an offspring $x'_i(G)$, by applying the crossover operator;
 if $f(x'_i(G))$ is better than $f(x_i(G))$ **then**
 Add $x'_i(G)$ to $P(G + 1)$;
 else
 Add $x_i(G)$ to $P(G + 1)$;
 end if
 end for
 Apply LFLS using Algorithm 2
end while
 Return the fittest solution;

Table 1. Test problems

| Test Problem | Objective function | Search Range | Optimum Value | D AE |
|-------------------------|---|---------------|-------------------------------|--------------|
| Sphere | $f_1(x) = \sum_{i=1}^D x_i^2$ | [-5.12, 5.12] | $f(0) = 0$ | 30 1.0E - 05 |
| De Jong f4 | $f_2(x) = \sum_{i=1}^D i \cdot (x_i)^4$ | [-5.12, 5.12] | $f(0) = 0$ | 30 1.0E - 05 |
| Griewank | $f_3(x) = 1 + \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}})$ | [-600, 600] | $f(0) = 0$ | 30 1.0E - 05 |
| Rosenbrock | $f_4(x) = \sum_{i=1}^D (100(x_{i+1} - x_i^2))^2 + (x_i - 1)^2$ | [-30, 30] | $f(1) = 0$ | 30 1.0E - 02 |
| Rastrigin | $f_5(x) = 10D + \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i)]$ | [-5.12, 5.12] | $f(0) = 0$ | 30 1.0E - 05 |
| Ackley | $f_6(x) = -20 + e + \exp(-\frac{0.2}{D} \sqrt{\sum_{i=1}^D x_i^3})$ | [-1, 1] | $f(0) = 0$ | 30 1.0E - 05 |
| Michalewicz | $f_7(x) = -\sum_{i=1}^D \sin x_i (\sin(\frac{x_i^2}{\pi}))^{20}$ | [0, π] | $f_{min} = -9.66015$ | 10 1.0E - 05 |
| Cosine Mixture | $f_8(x) = \sum_{i=1}^D x_i^2 - 0.1(\sum_{i=1}^D \cos 5\pi x_i) + 0.1D$ | [-1, 1] | $f(0) = -D \times 0.1$ | 30 1.0E - 05 |
| Exponential | $f_9(x) = -(\exp(-0.5 \sum_{i=1}^D x_i^2)) + 1$ | [-1, 1] | $f(0) = -1$ | 30 1.0E - 05 |
| Cigar | $f_{10}(x) = x_0^2 + 100000 \sum_{i=1}^D x_i^2$ | [-10, 10] | $f(0) = 04$ | 30 1.0E - 05 |
| Sum of different powers | $f_{11}(x) = \sum_{i=1}^D x_i ^{i+1}$ | [-1, 1] | $f(0) = 0$ | 30 1.0E - 05 |
| Step function | $f_{12}(x) = \sum_{i=1}^D (x_i + 0.5)^2$ | [-100, 100] | $f(-0.5 \leq x \leq 0.5) = 0$ | 30 1.0E - 06 |
| Inverted cosine wave | $f_{13}(x) = -\sum_{i=1}^{D-1} \left(\exp\left(\frac{-x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}}{8}\right) \times \mathbf{I} \right)$ | [-5, 5] | $f(0) = -D + 1$ | 10 1.0E - 05 |
| Levy montalvo 1 | where, $\mathbf{I} = \cos\left(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}}\right)$ $f_{14}(x) = \frac{\pi}{D} (10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2) \times (1 + 10 \sin^2(\pi y_{i+1})) + (y_D - 1)^2$, where $y_i = 1 + \frac{1}{4}(x_i + 1)$ | [-10, 10] | $f(-1) = 0$ | 30 1.0E - 05 |
| Levy montalvo 2 | $f_{15}(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2) \times (1 + \sin^2(3\pi x_{i+1})) + (x_D - 1)^2 (1 + \sin^2(2\pi x_D))$ | [-5, 5] | $f(1) = 0$ | 30 1.0E - 05 |

4.2 Experimental Setting

The basic DE (*DE/rand/1/bin*) [13] and its variants namely, Scale factor local search DE (*SFLSDE*) [11] and Fitness based DE (*FBDE*) [17] are compared with proposed LFDE algorithm and for the comparison, the adopted experimental setting is defined as follows:

- Population size $NP=50$ and the number of runs = 100.
- The scale factor $F=0.4$.
- For DE variants *SFLSDE* and *FBDE*, F and CR are set as suggested by their respective authors [14, 11].
- The stopping criteria is either achieving the mentioned acceptable error in table 1 or reaching the maximum number of function evaluations (which is set to be 2.0×10^5).
- The value of $\beta = 2$ is to be set based on the empirical experiments.
- To set ϵ i.e., the termination criteria of *LFDE*, different values of ϵ are analyzed to measure the performance of *LFDE* for considered test problems over 30 runs in Figure 1(a). Outcome in Figure 1(a) declares $\epsilon = 10$ as better choice as it gives better success rate. Therefore, termination criteria is set to be $\epsilon = 10$.
- To set the parameter CR , its sensitivity with respect to different values of CR in the range $[0.1, 1]$, is examined in the Figure 1(b) and the performance of *LFDE* is measured for 10 runs over 15 considered problems. It is observed from Figure 1(b) that the test problems are very sensitive towards CR and value 0.2 gives comparatively better success rate. Therefore CR is set to be 0.2.

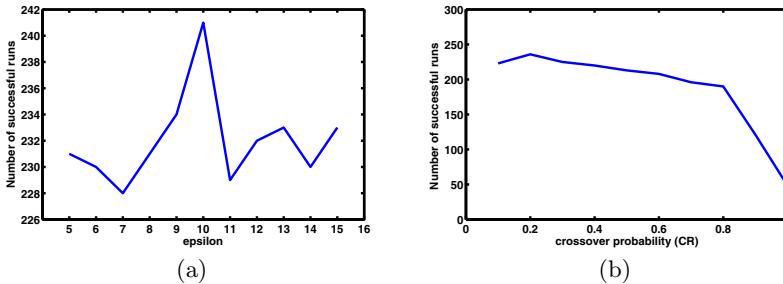


Fig. 1. (a) Effect of parameter ϵ on success rate, (b) Effect of parameter CR on success rate

4.3 Results Comparison

Table 2 displays the numerical outcomes of the proposed algorithm *LFDE* and basic *DE/rand/1/bin* and its variants *SFLSDE* and *FBDE* with experimental setting 4.2. Table 2 reported the performance in terms of four parameters namely, success rate (*SR*), average function evaluations (*AFEs*), mean error (*ME*) and standard deviation (*SD*). It can be observed from the table that

LFDE outperforms in terms of reliability, efficiency and accuracy as compare to the *DE/rand/1/bin*, *SFLSDE* and *FBDE* for most of the problems. Further, to analyze the performance of *LFDE* more deeply, acceleration rate (AR) [15], Mann-Whitney rank sum test [9] and success performance have been carried out for results of considered algorithms.

Table 2. Comparison of the results of test problems, TP: Test Problems

| TP | Algorithm | SD | ME | AFE | SR |
|----------|-----------|----------|----------|-----------|-----|
| f_1 | LFDE | 8.08E-07 | 9.09E-06 | 22467.45 | 100 |
| | DE | 8.24E-07 | 9.06E-06 | 22444 | 100 |
| | FBDE | 7.47E-07 | 9.22E-06 | 31923.83 | 100 |
| | SFLSDE | 8.65E-07 | 9.00E-06 | 24626.15 | 100 |
| f_2 | LFDE | 1.15E-06 | 8.61E-06 | 17942.3 | 100 |
| | DE | 8.51E-07 | 9.01E-06 | 20859.5 | 100 |
| | FBDE | 1.29E-06 | 8.42E-06 | 24369.23 | 100 |
| | SFLSDE | 1.17E-06 | 8.77E-06 | 20345.45 | 100 |
| f_3 | LFDE | 8.00E-07 | 9.20E-06 | 33253.7 | 100 |
| | DE | 5.00E-03 | 1.71E-03 | 55540 | 86 |
| | FBDE | 8.09E-07 | 9.16E-06 | 48091.22 | 100 |
| | SFLSDE | 1.03E-03 | 1.57E-04 | 39382.59 | 98 |
| f_4 | LFDE | 1.65E+01 | 3.35E+01 | 200000 | 0 |
| | DE | 4.78E+02 | 9.84E+01 | 200000 | 0 |
| | FBDE | 1.67E+01 | 3.11E+01 | 200000 | 0 |
| | SFLSDE | 2.93E+01 | 2.48E+01 | 199471.82 | 3 |
| f_5 | LFDE | 6.97E-07 | 9.18E-06 | 147122.32 | 100 |
| | DE | 4.63E+00 | 1.49E+01 | 200000 | 0 |
| | FBDE | 7.23E-07 | 9.22E-06 | 130754.62 | 100 |
| | SFLSDE | 7.83E-07 | 9.03E-06 | 153260.76 | 100 |
| f_6 | LFDE | 4.38E-07 | 9.51E-06 | 41936.39 | 100 |
| | DE | 4.42E-07 | 9.46E-06 | 43100.5 | 100 |
| | FBDE | 3.93E-07 | 9.59E-06 | 62374.9 | 100 |
| | SFLSDE | 4.19E-07 | 9.55E-06 | 46190.02 | 100 |
| f_7 | LFDE | 1.92E-06 | 7.83E-06 | 16734.33 | 100 |
| | DE | 4.26E-02 | 4.21E-02 | 171411 | 20 |
| | FBDE | 4.22E-03 | 5.72E-04 | 27415.76 | 96 |
| | SFLSDE | 3.19E-02 | 1.14E-02 | 66530.88 | 76 |
| f_8 | LFDE | 7.32E-07 | 9.12E-06 | 21677.2 | 100 |
| | DE | 4.72E-02 | 1.33E-02 | 37386 | 92 |
| | FBDE | 7.26E-07 | 9.15E-06 | 32219.5 | 100 |
| | SFLSDE | 5.77E-01 | 5.74E+00 | 197797.14 | 1 |
| f_9 | LFDE | 6.38E-07 | 9.10E-06 | 17054.85 | 100 |
| | DE | 9.15E-07 | 8.98E-06 | 17269 | 100 |
| | FBDE | 7.91E-07 | 9.16E-06 | 25087.79 | 100 |
| | SFLSDE | 7.78E-07 | 9.03E-06 | 23664.52 | 100 |
| f_{10} | LFDE | 8.38E-07 | 9.12E-06 | 39203.2 | 100 |
| | DE | 8.75E-07 | 9.02E-06 | 39861.5 | 100 |
| | FBDE | 8.97E-07 | 9.04E-06 | 54213.75 | 100 |
| | SFLSDE | 8.73E-07 | 8.97E-06 | 43232.55 | 100 |
| f_{11} | LFDE | 2.47E-06 | 7.03E-06 | 6613.17 | 100 |
| | DE | 2.06E-06 | 7.08E-06 | 7837 | 100 |
| | FBDE | 2.07E-06 | 7.40E-06 | 10695.75 | 100 |
| | SFLSDE | 1.99E-06 | 7.32E-06 | 8865.53 | 100 |
| f_{12} | LFDE | 2.38E-08 | 1.38E-07 | 14632.01 | 100 |
| | DE | 6.28E-01 | 1.90E-01 | 39465 | 87 |
| | FBDE | 2.92E-08 | 4.97E-07 | 20033.2 | 100 |
| | SFLSDE | 4.86E-08 | 6.78E-07 | 16309.22 | 100 |
| f_{13} | LFDE | 1.66E-06 | 8.53E-06 | 58968.34 | 100 |
| | DE | 6.55E-01 | 9.19E-01 | 171662 | 20 |
| | FBDE | 2.25E-06 | 7.58E-06 | 103044.78 | 100 |
| | SFLSDE | 8.24E-01 | 7.64E-01 | 132969.26 | 39 |
| f_{14} | LFDE | 7.49E-07 | 9.02E-06 | 20234.23 | 100 |
| | DE | 1.03E-02 | 1.05E-03 | 21505 | 99 |
| | FBDE | 7.92E-07 | 9.24E-06 | 29712.55 | 100 |
| | SFLSDE | 9.15E-07 | 8.87E-06 | 22139.07 | 100 |
| f_{15} | LFDE | 6.75E-07 | 9.05E-06 | 20342.31 | 100 |
| | DE | 1.87E-03 | 3.38E-04 | 25809 | 97 |
| | FBDE | 7.94E-07 | 9.16E-06 | 29314.37 | 100 |
| | SFLSDE | 8.05E-07 | 9.03E-06 | 24379.87 | 100 |

First, the considered algorithms are analyzed for the convergence speed by measuring the average function evaluations (AFE_s). A low value of AFE_s means higher convergence speed. Function evaluations for each problem is averaged over 100 simulations in order to minimize the effect of the stochastic nature of the algorithms. Based on AFE_s, acceleration rate (AR) is calculated to compare the convergence speed as:

$$AR = \frac{AFE_{ALGO}}{AFE_{LFDE}}, \tag{7}$$

where, ALGO is any of the considered algorithms with which we are comparing our proposed algorithm LFDE and AR > 1 means LFDE converges faster i.e., taking lesser number of function evaluations to converge. Table 3 shows convergence speed comparison between LFDE and DE, LFDE and FBDE, and LFDE and SFLSDE in terms of acceleration rate AR. It is clear from Table 3 that convergence speed of LFDE is higher than the other algorithms for most of the test problems.

Table 3. Acceleration Rate (AR) of LFDE compare to the basic DE, FBDE and SFLSDE, TP: Test Problems

| TP | DE | FBDE | SFLSDE |
|------------------------|-------------|-------------|-------------|
| <i>f</i> ₁ | 0.998956268 | 1.420892447 | 1.096081222 |
| <i>f</i> ₂ | 1.162587851 | 1.358199896 | 1.133937678 |
| <i>f</i> ₃ | 1.670190084 | 1.446191552 | 1.184307009 |
| <i>f</i> ₄ | 1 | 1 | 0.9973591 |
| <i>f</i> ₅ | 1.359413038 | 0.888747676 | 1.041723377 |
| <i>f</i> ₆ | 1.027758946 | 1.487369323 | 1.101430524 |
| <i>f</i> ₇ | 10.24307516 | 1.638294452 | 3.975712204 |
| <i>f</i> ₈ | 1.724669238 | 1.48633126 | 9.124662779 |
| <i>f</i> ₉ | 1.012556545 | 1.471006195 | 1.387553687 |
| <i>f</i> ₁₀ | 1.016791997 | 1.382890937 | 1.102781151 |
| <i>f</i> ₁₁ | 1.18505951 | 1.617340852 | 1.340587041 |
| <i>f</i> ₁₂ | 2.697168742 | 1.369135204 | 1.114626083 |
| <i>f</i> ₁₃ | 2.911087543 | 1.747459399 | 2.254926288 |
| <i>f</i> ₁₄ | 1.062802983 | 1.468429982 | 1.094139485 |
| <i>f</i> ₁₅ | 1.268734967 | 1.441054138 | 1.1984809 |

Further, the Mann-Whitney rank sum [9], a non-parametric test, which is a well established test for comparison among non-Gaussian data, is applied for the comparison of the considered algorithms. In this paper, this test is performed at 5% level of significance (α = 0.05) between LFDE - DE, LFDE - FBDE and LFDE - SFLSDE. Table 4 shows the results of the Mann-Whitney rank sum test for the average number of function evaluations of 100 simulations. First we observed the significant difference by Mann-Whitney rank sum test i.e., whether the two data sets are significantly different or not. If significant difference is not seen (i.e., the null hypothesis is accepted) then sign '=' appears and when significant difference is observed i.e., the null hypothesis is rejected then compare the average number of function evaluations. And we use signs '+' and '-' for the case where LFDE takes less or more average number of function

Table 4. Results based on mean function evaluations and the Mann-Whitney rank sum test at $\alpha = 0.05$ significance level, TP: Test Problems

| TP | LFDE Vs DE | LFDE Vs FBDE | LFDE Vs SFLSDE |
|---------------------------|------------------|--------------------|----------------------|
| f_1 | = | + | + |
| f_2 | + | + | + |
| f_3 | + | + | + |
| f_4 | = | = | = |
| f_5 | + | - | + |
| f_6 | + | + | + |
| f_7 | + | + | + |
| f_8 | + | + | + |
| f_9 | + | + | + |
| f_{10} | + | + | + |
| f_{11} | + | + | + |
| f_{12} | + | + | + |
| f_{13} | + | + | + |
| f_{14} | + | + | + |
| f_{15} | - | + | + |
| Total number of + sign | 12 | 13 | 14 |

Table 5. Comparison among *LFDE*, *DE*, *FBDE* and *SFLSDE* based on Success Performance (SP) for scalable problems, TP: Test Problems

| TP | LFDE | DE | FBDE | SFLSDE |
|----------|-----------|-------------|-------------|-------------|
| f_1 | 22467.45 | 22444 | 31923.83 | 24626.15 |
| f_2 | 17942.3 | 20859.5 | 24369.23 | 20345.45 |
| f_3 | 33253.7 | 64581.39535 | 48091.22 | 40186.31633 |
| f_4 | 1E+24 | 1E+24 | 1E+24 | 6666666.667 |
| f_5 | 147122.32 | 1E+24 | 130754.62 | 153260.76 |
| f_6 | 41936.39 | 43100.5 | 62374.9 | 46190.02 |
| f_7 | 16734.33 | 857055 | 28558.08333 | 87540.63158 |
| f_8 | 21677.2 | 40636.95652 | 32219.5 | 19779714 |
| f_9 | 17054.85 | 17269 | 25087.79 | 23664.52 |
| f_{10} | 39203.2 | 39861.5 | 54213.75 | 43232.55 |
| f_{11} | 6613.17 | 7837 | 10695.75 | 8865.53 |
| f_{12} | 14632.01 | 45362.06897 | 20033.2 | 16309.22 |
| f_{13} | 58968.34 | 858310 | 103044.78 | 340946.8205 |
| f_{14} | 20234.23 | 21722.22222 | 29712.55 | 22139.07 |
| f_{15} | 20342.31 | 26607.21649 | 29314.37 | 24379.87 |

evaluations than the other’s, respectively. Therefore in Table 4, ‘+’ shows that LFDE is significantly better and ‘-’ shows that LFDE is worse. As Table 4 includes 39 ‘+’ signs out of 45 comparisons, it can be concluded that LFDE performs significantly better than DE, FBDE and SFLSDE.

Success Performance (*SP*) [7] of all the considered algorithms is also quantified in the paper using the equation:

$$SP = \frac{\text{AFEs}}{\text{Number of Successful Runs}} \times \text{Total Runs}, \tag{8}$$

The calculated *SPs* of all the considered algorithms are shown in Table 5 and corresponding box plots are given in Figure 5. Any algorithm seems to be good if it takes less function evaluations and more successful runs, so it is clear from

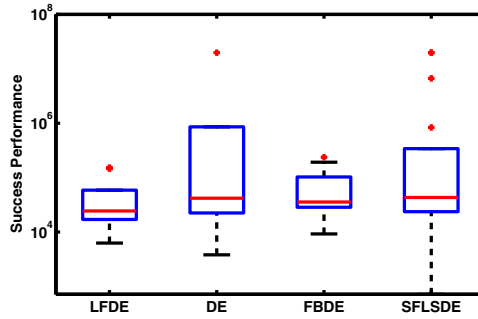


Fig. 2. LFDE:Boxplots for Success Performance

equation (8) that algorithm having low SP will be considered better. If none success is achieved by any algorithm then we assigned a large number 6×10^4 as success performance to that problem so that it will be called worst as having high SP value. It is clear from Figure 2 that $LFDE$ is better among the considered algorithms as interquartile range and median of SP are comparatively low.

5 Conclusion

This paper incorporated a Levy Flight based local search strategy in DE. Proposed strategy ($LFDE$) enables the algorithm to generate new solutions in the vicinity of the best solution based on parameter namely perturbation rate i.e., enables exploitation of the search space in the neighborhood of current population's best candidate solution. Further, the proposed algorithm is tested on 15 optimization problems and compared to basic DE and its recent variants namely, Fitness based DE ($FBDE$) and Scale Factor Local Search DE ($SFLSDE$). Different Statistical analysis is done in the paper to prove the reliability, efficiency and accuracy of proposed algorithm $LFDE$ over the other algorithms.

References

1. Ali, M.M., Khompatraporn, C., Zabinsky, Z.B.: A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of Global Optimization* 31(4), 635–672 (2005)
2. Chakraborty, U.K.: *Advances in differential evolution*. Springer (2008)
3. Das, S., Konar, A.: Two-dimensional iir filter design with modern search heuristics: A comparative study. *International Journal of Computational Intelligence and Applications* 6(3), 329–355 (2006)
4. Holland, J.H.: *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor (1975)
5. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948. IEEE (1995)

6. Lampinen, J., Zelinka, I.: On stagnation of the differential evolution algorithm. In: Proceedings of MENDEL, pp. 76–83 (2000)
7. Liang, J.J., Philip Runarsson, T., Mezura-Montes, E., Clerc, M., Suganthan, P.N., Coello Coello, C.A., Deb, K.: Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. *Journal of Applied Mechanics* 41 (2006)
8. Liu, P.K., Wang, F.S.: Inverse problems of biological systems using multi-objective optimization. *Journal of the Chinese Institute of Chemical Engineers* 39(5), 399–406 (2008)
9. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics* 18(1), 50–60 (1947)
10. Mezura-Montes, E., Velázquez-Reyes, J., Coello Coello, C.A.: A comparative study of differential evolution variants for global optimization. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 485–492. ACM (2006)
11. Neri, F., Tirronen, V.: Scale factor local search in differential evolution. *Memetic Computing* 1(2), 153–171 (2009)
12. Omran, M.G.H., Engelbrecht, A.P., Salman, A.: Differential evolution methods for unsupervised image classification. In: The 2005 IEEE Congress on Evolutionary Computation, vol. 2, pp. 966–973. IEEE (2005)
13. Price, K.V.: Differential evolution: a fast and simple numerical optimizer. In: 1996 Biennial Conference of the North American Fuzzy Information Processing Society, NAFIPS, pp. 524–527. IEEE (1996)
14. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: The 2005 IEEE Congress on Evolutionary Computation, vol. 2, pp. 1785–1791. IEEE (2005)
15. Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.A.: Opposition-based differential evolution. *IEEE Transactions on Evolutionary Computation* 12(1), 64–79 (2008)
16. Rogalsky, T., Kocabiyik, S., Derksen, R.W.: Differential evolution in aerodynamic optimization. *Canadian Aeronautics and Space Journal* 46(4), 183–190 (2000)
17. Sharma, H., Bansal, J.C., Arya, K.V.: Fitness based differential evolution. *Memetic Computing* 4(4), 303–316 (2012)
18. Sharma, H., Bansal, J.C., Arya, K.V.: Opposition based lévy flight artificial bee colony. *Memetic Computing*, 1–15 (2012)
19. 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)
20. 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, CEC 2004, vol. 2, pp. 1980–1987. IEEE (2004)
21. Yang, X.S.: Nature-inspired metaheuristic algorithms. Luniver Press (2010)

An Adaptive Differential Evolution Based Fuzzy Approach for Edge Detection in Color and Grayscale Images

Satrajit Mukherjee¹, Bodhisattwa Prasad Majumder¹,
Aritran Piplai², and Swagatam Das³

¹ Dept. of Electronics and Telecommunication Engg.,
Jadavpur University, Kolkata 700 032, India

² Dept. of Computer Science and Engg., Jadavpur University, Kolkata 700 032, India

³ Electronics and Communication Sciences Unit,
Indian Statistical Institute, Kolkata – 700 108, India
{satra0293,mbodhisattwa}@gmail.com, mpiplai@yahoo.com,
swagatam.das@ieee.org

Abstract. This paper presents a novel optimal edge detection scheme based on the concepts of fuzzy Smallest Univalued Assimilating Nucleus (SUSAN) and JADE (an adaptive Differential Evolution variant with Optional External Archive). Initially, the Univalued Assimilating Nucleus (USAN) area is calculated from the intensities of every neighborhood pixel of a pixel of interest in the test image. The USAN area edge map of each of the RGB components is fuzzified subject to the optimization of fuzzy entropy, together with fuzzy edge sharpness factor, using JADE. Adaptive thresholding converts the fuzzy edge map to spatial domain edge map. Finally, the individual RGB edge maps are concatenated to obtain the final image edge map. Qualitative and quantitative comparisons have been rendered with respect to a few promising edge detectors and also optimal fuzzy edge detectors based on metaheuristic algorithms like the classic Differential Evolution (the classic DE/rand/1) and the Particle Swarm Optimization (PSO).

1 Introduction

Edges of an image are mainly boundaries between two regions which have great difference in their respective pixel intensities. Edges help to demarcate an image into objects and background.

Edge detectors can be broadly classified into directional [1-3] and non-directional edge detectors [4, 5]. However, every existing edge detector has its own set of drawbacks. The Canny edge detection [4, 5] algorithm, for instance, does not guarantee that the intensity discontinuities in an image correspond to actual edges of the object. Future modifications of the Canny algorithm include Rothwell [6] (Canny with added topology) and Edison [7] (modified Canny with added measure of confidence and use of templates). These detectors show improved performance over

Canny in detecting junction edges as well as edges in low contrast regions. Roberts Cross Edge detector [8] takes very less computational time since only four input pixels need to be mathematically operated upon to determine the value of each output pixel. However, since it uses a very small kernel, it produces very weak responses to genuine edges unless they are in high contrast zones. Sobel [9] detector performs better than Roberts in this aspect as it uses a 3x3 instead of a 2x2 kernel and thus produces higher output values for similar edges than the latter. However, the smoothing effect of this operator leads to edges that are several pixels wide which is undesirable. Zero-crossing based edge detectors like the Laplacian of the Gaussian (LoG) (based on the Marr-Hildreth operator) [10] test wider areas around the pixels and find correct places of edges, both weak and strong, but malfunction at corners and curves where the gray level intensity function varies. As the LoG detector strictly uses the laplacian filter, it fails to find the correct orientation of edges at such corners and curves. Non-gradient edge detectors like the SUSAN [11] detect edges in the image by associating a small area of neighbourhood pixels of similar intensity to each centre pixel. The SUSAN principle is widely applied in corner detection [12] and works better than LoG in finding the proper orientation of edges. However, the SUSAN is unable to detect weak edges and suffers from poor edge connectivity. Additionally, it produces thick blurry edges which require sharpening. A recently developed algorithm based on the Bacterial Foraging Optimization [13] aims to extract edge information by driving bacteria along the direction where probability of finding nutrients (edge) is maximum. But the performance of this algorithm is governed by several parameters like swim length, chemotactic steps, reproduction steps, elimination-dispersion steps, etc. whose simultaneous tuning for optimal test results is quite difficult. The parameter set empirically chosen in [13] yields disconnected edges due to restricted swim length and parallel edges due to path retracing by the bacteria.

In our proposed algorithm, information about the relatively weaker edges is extracted from the SUSAN based USAN area edge map, using fuzzy measures. The parameters controlling the range and shape of the proposed bell-shaped curve, used for detecting the weaker edges, are determined by optimizing the fuzziness or the fuzzy entropy of the bell-shaped membership function. The concept of sharpness factor of an edge map is introduced so that sharpening of the SUSAN based USAN area edges are possible by regulating this factor. While the Shannon Entropy [14] is a measure of the randomness associated with a random variable, the fuzzy entropy [15] realizes the concept of entropy by using fuzzy sets, thus, containing possibilistic uncertainties instead of probabilistic uncertainties. Hence, we choose to optimize the fuzzy entropy value instead of the Shannon entropy, together with the sharpness factor of the image so that not only all the essential edge information is extracted but adequate sharpness of the edge map is also attained. Our edge detection approach employs an adaptive differential evolution with optional external archive [24], an efficient mutation strategy also known as JADE (named after the initials of its authors) to achieve this optimization. JADE has found widespread applications in several optimization problems, where it has been found to show improved performance from the classic DE as well as other search methods.

The organization of the paper is as follows: Section 2.1 introduces the SUSAN principle for the calculation of USAN area. Section 2.2 describes the fuzzification of USAN area using fuzzy membership functions while section 2.3 introduces the fuzzy measures required to evaluate the fuzzy entropy and sharpness factor of the fuzzy edge map. The need for adaptive thresholding is emphasized in section 2.4. Section 3 presents the objective function required for optimization. Section 4 presents the experimental results along with their qualitative and quantitative analyses while Section 5 concludes the proceedings. The quantitative measures include Kappa Value and Pratt’s Figure of Merit. As is evident from the experimental results, the performance of the JADE based fuzzy edge detector remains commendable for both colour and gray scale images.

1.1 Computation of SUSAN Based USAN Area

SUSAN [11] area principle has been invoked in computing the USAN area of the circular mask shown in Fig.1. The gray levels of the neighbouring 36 pixels are compared with the intensity of the centre pixel using the comparing equation and the outputs are summed up to obtain the USAN area. The concept of each image pixel, having associated with it a local (USAN) area of similar brightness, is the basis for the SUSAN principle.

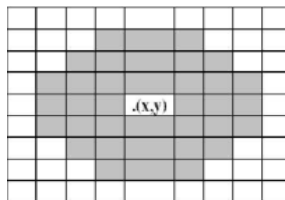


Fig. 1. Circular Mask (orientation 3,5,7,7,7,5,3 pixels from top to bottom)

The comparison equation of any pixel with the centre pixel is as follows:

$$F(r, r_0) = \exp \left[\left(\frac{(-I(r) - I(r_0))}{t} \right)^6 \right] \tag{1}$$

where ‘ r ’ is the position of any pixel, ‘ r_0 ’ is the position of the centre pixel, $I(r)$ is the intensity of any pixel, $I(r_0)$ is the intensity of the centre pixel and ‘ t ’ is the threshold brightness, empirically chosen as 20 for all test cases.

Eq. (1) is applied on the mask and then the sum of the outputs is computed from the equation:

$$U(r_0) = \sum_r F(r, r_0) \tag{2}$$

This $U(r_0)$ gives the USAN area of the centre pixel at the position and may vary from 1 to 37. Thus the USAN area serves as a measure of the number of pixels having the same gray level as that of the centre pixel.

1.2 Fuzzification of Usan Area

The probability of an USAN area with a particular number of pixels is calculated as below:

$$p(k) = h(k) / \sum h \tag{3}$$

where ‘ k ’ varies from 1 to 37 i.e. the possible range of values of the USAN area, $p(k)$ is the frequency of occurrence of USAN area k and $h(k)$ is the total number of pixels having area k , with $p(k)$ satisfying the condition

$$\sum_0^{L-1} p(k) = 1, \tag{4}$$

where ‘ L ’ is the total number of pixels in the circular mask, i.e. in this case 37.

Fuzzy memberships indicate how strongly a gray level belongs to the background or to the foreground. These membership functions convert the intensity values in the spatial domain into membership values of the range 0 to 1 and here it is done with the help of Eq. (5) which represents a histogram based Gaussian Membership Function:

$$\mu_1(k) = \exp\left(-\left(\frac{(x_{max}-k)^2}{2 * f_h^2}\right)\right). \tag{5}$$

The fuzzifier parameter, ‘ f_h ’ is calculated in the following way:

$$f_h^2 = \frac{\sum_0^{L-1} (x_{max}-k)^4 * p(k)}{2 * \sum_0^{L-1} (x_{max}-k)^2 * p(k)}. \tag{6}$$

Here ‘ k ’ is the USAN area, ‘ x_{max} ’ is the maximum USAN area. L here is 37 as stated before. A bell shaped function meant to enhance the membership function values of the original gray level values of the image is given by:

$$\mu_2(k) = \frac{1}{\left(1 + \frac{|\mu_1(k) - a|^{2b}}{c}\right)}, \tag{7}$$

where ‘ a ’, ‘ b ’, ‘ c ’ can be changed as per the user, which makes it more flexible. Weak edges are mainly detected because of this bell-shaped function which acts as a hedge.

1.3 Fuzzy Measures

Fuzzy Edge Sharpness Measures

Fuzzy edge sharpness measures [16-24] help to determine the quality of the different edges, thereby serving as a measure by which strong edges can be distinguished from weak edges.

The fuzzy edge sharpness factor for the weak edges can be formulated as:

$$F_w = \frac{1}{L} \sum_0^{L-1} (\mu_2(k) - c)^2 * p(k). \tag{8}$$

The average fuzzy edge sharpness factor for the weak edges can be formulated as:

$$F_{avgW} = \frac{1}{L} \sum_0^{L-1} (\mu_2(k) - c) * p(k). \tag{9}$$

The fuzzy edge sharpness factor for the strong edges can be formulated as:

$$F_s = \frac{1}{L} \sum_0^{L-1} (\mu_1(k) - c)^2 * p(k). \quad (10)$$

The average sharpness factor for the strong edges can be formulated as:

$$F_{avgS} = \frac{1}{L} \sum_0^{L-1} (\mu_1(k) - c) * p(k). \quad (11)$$

The average fuzzy edge sharpness factors indicate the overall sharpness of an image, whereas the individual fuzzy sharpness factors determine the actual deviations from the crossover point (here denoted by c).

A fuzzy edge quality factor is formed by taking the ratio of the average to the actual value of the fuzzy edge sharpness factors. This quality factor determines the uncertainty present in the edges.

The fuzzy edge quality factor for weak edges is denoted by:

$$Q_w = \frac{F_{avgW}}{F_w} \quad (12)$$

The fuzzy edge quality factor for strong edges is denoted by:

$$Q_s = \frac{F_{avgS}}{F_s} \quad (13)$$

Fuzzy Entropy

Fuzzy Entropy [15] serves as a measure of the uncertainty or the randomness associated with the image information (the image being the edge map in this case) and is defined by:

$$E = \frac{-1}{L * \ln 2} \sum_0^{L-1} \left[\mu_2(k) * \ln(\mu_2(k)) + (1 - \mu_2(k)) * \ln(1 - \mu_2(k)) \right] \quad (14)$$

Fuzzy entropy, in this case, serves as a measure of the amount of information that can be extracted from an image and its optimization yields the optimal parameter values of the bell-shaped membership function.

Sharpness Factor

The normalized fuzzy edge sharpness factor or the sharpness factor is required to calculate the fuzzy edge sharpness. The sharpness factor is defined as:

$$S_f = \frac{Q_s}{Q_w} \quad (15)$$

This definition helps us to fix a limit on the fuzzy edge sharpness factor, because increasing its value beyond the limit would cause loss of edge information. For a good edge map, the value of the sharpness factor should be in the range 1 to 1.5.

1.4 Adaptive Thresholding

Image thresholding [28-30], be it global or local, takes a grayscale image as input and produces a binary image as output. Global thresholding processes compute a global threshold value for the entire grayscale image which demarcates between the foreground and the background pixels. However, an image may have spatial variations in illumination and in such cases, the threshold has to be dynamically

varied over the image. There are various processes of adaptive thresholding for eg. the Chow and Kaneko Method [29], local thresholding [30], etc. The Chow and Kaneko method divides the image into a number of sub-images and calculates the threshold for the individual sub-images. Finally it interpolates the results of the sub-images to determine the threshold for each pixel. Local thresholding (as has been applied in our algorithm) investigates the neighborhood of each pixel and statistically determines the threshold for each pixel. The statistic used may be mean, median or mean of the maximum and minimum intensity values in the neighborhood. The statistic used in thresholding the fuzzy domain edge maps to obtain the final edge maps in our case was mean and the window size was kept between 7x7 to 15x15, depending on image size.

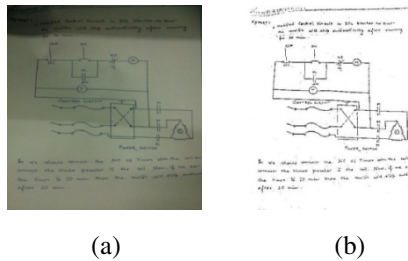


Fig. 2. a) An image having changing lighting conditions b) Image obtained after application of local adaptive thresholding

2 Objective Function For Optimization

Fuzzy entropy is a function of fuzzy sets that becomes lesser when the sharpness of the argument fuzzy set is improved. The entropy as well as the edge sharpness factor determines edge quality. So their simultaneous optimization is necessary.

The entropy function E is optimized subject to the constraint

$$S_{df} = S_f \tag{16}$$

For this, an objective function J is defined as:

$$J = E + \lambda * |S_{df} - S_f| \tag{17}$$

The value of λ is chosen to be 0.5 and S_{df} is chosen as 1.5 to yield a desirably sharp edge map. While optimizing J several experimentally found constraints were applied after which the set of constraints $-4 < a < 0$; $1 < b < 6$; $1 < c < 4$ was chosen. Optimization was carried out using JADE, where the a, b, c parameters of the chosen 30 member population were initialised subject to the set of constraints presented above. The fuzzifiers f_h for the USAN area edge maps of all the test images, calculated using Eq. (6), prior to the optimization of the objective function J , were found to be in the range of 15 to 22. A low value of f_h , generally below 5, corresponds to extremely thick edges while a high value, typically beyond 60, leads to loss of information due to high S_f . Thus the f_h values of all the population members were initialized within 5 to 60.

3 Experimental Results

The proposed approach has been evaluated on well-known test images of varying levels of complexities and its performance has been analyzed with respect to existing as well as state-of-the-art edge detection schemes. Fig. 3 presents the five test images while Fig. 4 shows the final RGB edge map and the individual RGB edge maps of the test image Lena. Table 1 shows the optimal values of the parameters a , b , c and f_h for the RGB edge maps.

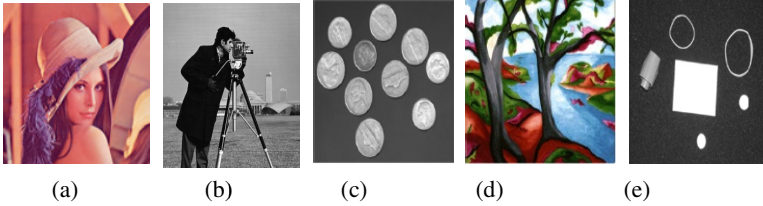


Fig. 3. (a) Lena (b) Cameraman (c) Coins (d) Trees (e) Pillsetc

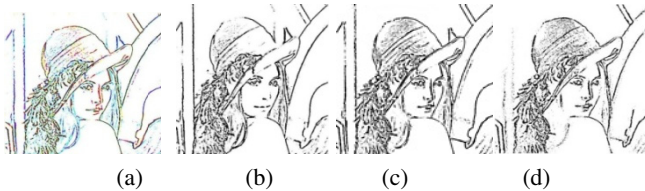


Fig. 4. (a) Final RGB edge map (Lena) (b) R component edge map (c) G component edge map (d) B component edge map

Table 1. RGB Optimized Values(Using JADE) For Lena

| Parameters | RED | GREEN | BLUE |
|------------|---------|---------|---------|
| a | -1.9999 | -1.9999 | -1.9999 |
| b | 3.0006 | 3.0004 | 3.0011 |
| c | 2.2763 | 2.2583 | 2.2951 |
| f_h | 23.6190 | 23.1441 | 24.6299 |

3.1 Visual Comparison with Existing Edge_Detectors

Figures 5 and 6 qualitatively compare our binarized edge maps (for two of the five test images), to the edge maps obtained using traditional as well as recent edge detectors, mentioned previously in Section 1. The proposed algorithm was also implemented with the DE/rand/1 [25] as well as the Particle Swarm Optimizer [31], (parameters chosen from their standard literature) and the results are analyzed qualitatively in this subsection and quantitatively in the next. Note that the ground truth image was formed using the Canny, Edison, Rothwell, LoG, Roberts, Sobel and SUSAN edge maps.

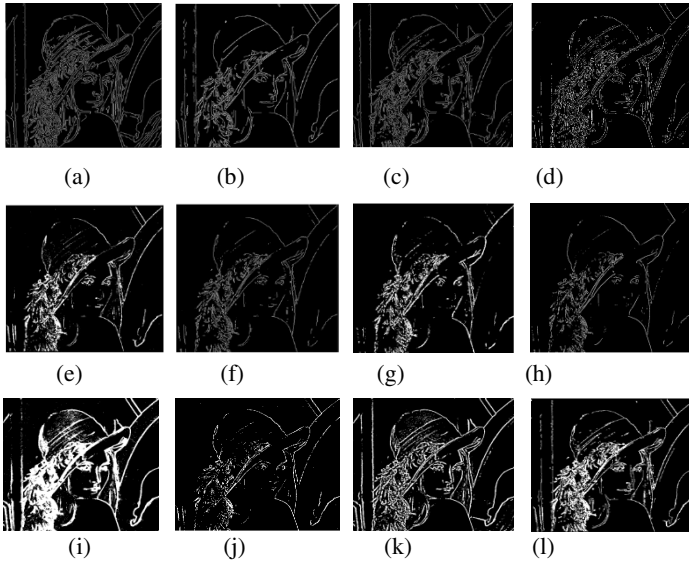


Fig. 5. (a) Canny(b) Edison (c) Rothwell (d) LoG (e) SUSAN (f) Roberts (g) Sobel (h) BFA Based Edge Detector (i) Optimal Fuzzy(DE) (j) Optimal Fuzzy(PSO) (k) Proposed Method (Optimal Fuzzy(JADE)) (l) True/ Majority Image

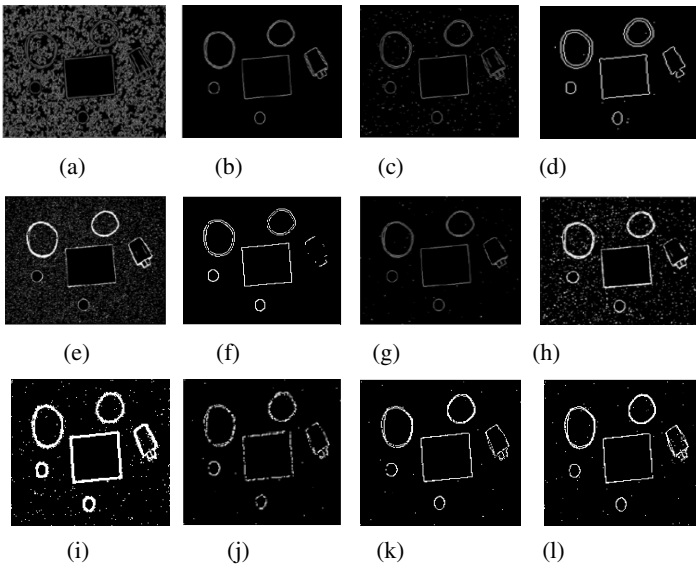


Fig. 6. (a) Canny (b) Edison(c) Rothwell (d) LoG (e) SUSAN (f) Roberts (g) Sobel (h) BFA based Detector (i) Optimal Fuzzy(DE) (j) Optimal Fuzzy(PSO) (k) Proposed Method (Optimal Fuzzy(JADE)) (l) True/ Majority Image

Figures 5 & 6 indicate that the optimal fuzzy edge detector using JADE performs better than the other edge detectors. For instance in **Fig. 5**, the weak edges to the rightmost bottom corner, the leftmost upright structure, etc. are almost indeterminable in the **PSO (Fig. 5(j))** and **BFA based systems (Fig. 5(h))** while they are almost clearly visible in the **fuzzy JADE edge map (Fig. 5(k))**. Additionally, the PSO and BFA based spatial edge maps show weak edge connectivity and inability to detect the weak edges completely. It is observed that though the edges are accurately detected using the BFA based detector [13], the method lacks in presenting complete edges due to restrictions imposed on the maximum swim length for a bacterium. Also, thick edges appear due to bacteria moving parallel to an edge. On the other hand, **the DE/rand/1-based spatial domain edge maps (Figures 5(i), 6(i))** show inadequate sharpening of edges and hence this method shows no significant improvement over the **SUSAN (Figures 5(e), 6(e)) method of edge detection** which also suffers from the problem of having thick and blurry edges. As is evident from the experimental results, the performance of the JADE based fuzzy edge detector remains commendable for all the test images and as compared to the other metaheuristic based fuzzy and non-fuzzy algorithms.

As for the traditional edge detectors, **Canny (Figs. 5(a), 6(a))** and **Rothwell (Figs. 5(c), 6(c))** edge detectors produce double edges while **Edison (Figs. 5(b), 6(b))** shows abrupt discontinuities in the edge maps. **Roberts (Figs. 5(f), 6(f))** and **Sobel (Figs. 5(g), 6(g))** detect minimum number of edges among all these edge detectors due to restricted kernel size. **LoG edge maps ((Figs. 5(d), 6(d))** malfunction at corners. The optimization of fuzzy entropy using JADE assigns optimal membership values to both weak and strong edges in such a way that even the relatively weaker edges in the low contrast regions as well as in poorly illuminated regions can be distinguished. The edge quality is also improved by regulating the image sharpness factor. Hence our proposed edge detection approach preserves the unique characteristics of each image without distorting it and also serves as a good detector of weak edges.

3.2 Quantitative Analysis

Kappa Value

Kappa value [32] of an image with respect to a ground truth or majority image is a measure of its pixel-to-pixel similarity with the truth image. A majority or truth image in our case has been formed from the edge maps obtained by using Canny, Edison, Rothwell, LoG, Roberts, Sobel and SUSAN. A ground truth pixel will be an edge pixel if majority of the edge detectors mentioned above produce an edge pixel at that coordinate. Table 2 shows the Kappa Values of all the edge detectors. Our proposed approach shows maximum truthfulness or maximum agreement with the majority image as is evident from its highest kappa values for all test images. Kappa Value is a statistic given by:

$$Kappa = \frac{Observed\ Agreement - Expected\ Agreement}{1 - Expected\ Agreement} \quad (18)$$

Table 2. Kappa Values for various edge maps

| ALGORITHM | KAPPA VALUES | | | | |
|--------------------|---------------|---------------|---------------|---------------|---------------|
| | Lena | Cameraman | Coins | Trees | Pillsetc |
| Canny | 0.4592 | 0.4174 | 0.5543 | 0.5343 | 0.2243 |
| Edison | 0.4654 | 0.4364 | 0.3654 | 0.5254 | 0.5754 |
| Rothwell | 0.4384 | 0.5284 | 0.4984 | 0.5374 | 0.5941 |
| LoG | 0.3872 | 0.4974 | 0.5011 | 0.5115 | 0.5621 |
| SUSAN | 0.4250 | 0.5564 | 0.4680 | 0.5184 | 0.4380 |
| Roberts | 0.3441 | 0.3104 | 0.3291 | 0.3971 | 0.4142 |
| Sobel | 0.3792 | 0.3242 | 0.3743 | 0.4102 | 0.4292 |
| BF based system | 0.5125 | 0.6393 | 0.5125 | 0.5784 | 0.5255 |
| Fuzzy(PSO) | 0.4997 | 0.3167 | 0.4212 | 0.4307 | 0.4017 |
| Fuzzy(DE) | 0.5433 | 0.6403 | 0.4733 | 0.5641 | 0.5134 |
| Fuzzy(JADE) | 0.6401 | 0.6899 | 0.6594 | 0.6112 | 0.6971 |

Pratt’s Figure of Merit

Pratt’s Figure of Merit (FOM) [33] measures the ideality factor of an edge detector on a scale of 0 to 1, based on its ability to detect and localize edge points and distinguish noisy pixels from edge pixels. An FOM value of 1 indicates that the edge map produced by the algorithm perfectly resembles the ideal edge map whereas an FOM value of 0 refers to complete deviation from ideality. FOM is mathematically represented as:

$$FOM = \frac{1}{I_N} \sum_1^{I_A} \frac{1}{1+f*d^2}, \tag{19}$$

where $I_N = \max(I_A, I_I)$, I_A represents the total number of edge pixels in the obtained edge map while I_I refers to the edge pixels in the ideal edge map. The scaling factor f is kept at a value of 0.9 while d represents the distance between the actual edge pixel and the nearest ideal pixel. The normalization is done with the maximum of I_A and I_I to take into consideration the penalty that may occur due to smeared ($I_A > I_I$) or fragmented edges ($I_A < I_I$).

In Figures 8 and 9, our proposed method is compared with three of the existing traditional edge detectors, search-based Canny and Sobel and zero-crossing based LoG, and also with the recently developed bacterial foraging based edge detector [13] and optimal fuzzy edge detection schemes based on PSO and DE/rand/1. The Pratt values of the edge maps are computed and compared in Table 3 with two sample edge maps taken from the Berkeley Segmentation Dataset (<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds>), shown in Fig. 7(a) and 7(b), serving as the ground truth images.

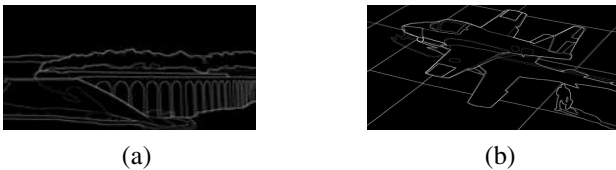


Fig. 7. (a) Truth Image (Test Image 1) (b) Truth Image (Test Image 2)

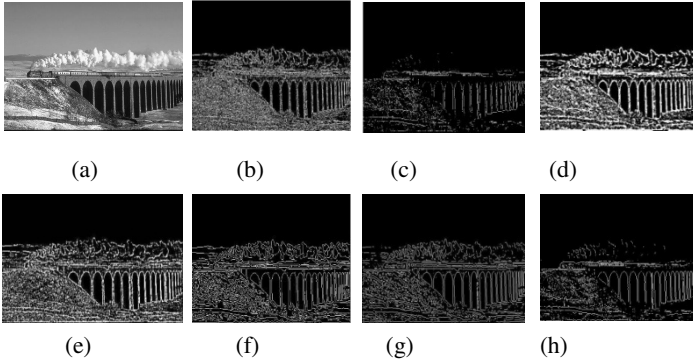


Fig. 8. (a) Test Image 1 (b) Proposed Approach (using JADE) (c) Optimal Fuzzy (PSO) edge map (d) Optimal Fuzzy (DE) edge map (e) Bacterial Foraging based edge map (f) Canny Edge Map (g) LoG Edge Map (h) Sobel Edge Map

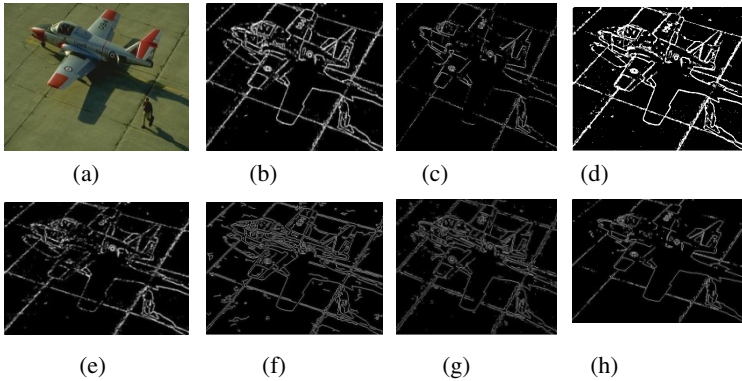


Fig. 9. (a) Test Image 2 (b) Proposed Approach (using JADE) (c) Optimal Fuzzy (PSO) (d) Optimal Fuzzy (DE) edge map (e) BF based edge detector (f) Canny Edge Map (g) LoG Edge Map (h) Sobel Edge Map

Table 3. Pratt's Figure Of Merit

| Algorithm | Pratt's Figure Of Merit | |
|----------------------------|-------------------------|---------------|
| | Test Image 1 | Test Image 2 |
| Canny | 0.6022 | 0.5232 |
| LoG | 0.5631 | 0.4458 |
| Sobel | 0.3156 | 0.3080 |
| BF based system | 0.5784 | 0.4958 |
| Optimal Fuzzy(DE) | 0.6232 | 0.5771 |
| Optimal Fuzzy(PSO) | 0.5883 | 0.5536 |
| Optimal Fuzzy(JADE) | 0.6753 | 0.6635 |

Table 3 shows that in both cases, our proposed method yields the highest value of FOM, thus behaving as the closest approximate to the ideal edge maps. Canny and LoG produce too many double edges and hence have low FOM value. Sobel yields a very low FOM value as it detects lesser number of edges. While the DE based system suffers from thick edges due to inadequate sharpening, the BFA and PSO based systems show poor edge connectivity and inability to detect weak edges, along with fat blurry edges in case of the BFA based system. The optimal fuzzy edge maps (using JADE) have regulated sharpness and also show proper detection of weak edges, thus justifying the relatively high FOM or Pratt values of the edge maps.

Note that the quantitative measures for all metaheuristic algorithms were calculated after averaging through 25 independent runs for each test image, each run being continued till 60,000 function evaluations.

4 Conclusions

The experimental results show that the proposed optimal fuzzy edge detection algorithm, using an adaptive DE with optional external archive (popularly known as JADE), is able to detect weak edges accurately by optimizing the parameters that control the shape of the bell-shaped fuzzy membership function. Bi-level adaptive thresholding binarizes the fuzzy edge maps to yield the spatial domain edge maps for the RGB components in case of a colour image. The main advantage of using this method is that:

- a) Our edge detector retains the shape of the image, shows great edge connectivity and has improved edge quality, as indicated by the relatively high Pratt's figure of merit [33] and optimal Shannon's Entropy [14] value.
- b) Owing to the ability of this edge detector to detect weak edges accurately while still preserving the shape of the object, it can be used in biometric applications like palm print identification, face detection, contour feature extraction, etc.

Future work with the proposed approach would be to extend it for noisy images and images subjected to motion blur. The main disadvantage of this method is that it fails to suppress edges due to substantial amount of noise that distorts the shape of the image.

References

1. Beucher, S., Meyer, F.: The morphological approach to segmentation: The Watershed transformation. In: Dougherty, E.R. (ed.) *Mathematical Morphology in Image Processing* (1993)
2. Kovacevic, J., Vetterli, M.: Non separable multidimensional perfect reconstruction filter banks and wavelet for Rn. *IEEE Transactions on Information Theory* 38(2), 533–555 (1992)
3. Mallat, S., Zhong, S.: Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(7), 710–732

4. Canny, J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8(1), 679–698 (1986)
5. Yao, Y., Ju, H.: A Sub - pixel edge detection method based on Canny operator. In: *Proceedings of 6th International Conference on Fuzzy Systems and Knowledge Discovery*, Tianjin, August 16-18, pp. 97–100 (2009)
6. Rothwell, C.A., Mundy, J.L., Hoffman, W., Nguyen, V.: *Driving Vision by Topology*. In: *TR-2444 – Programme 4*, pp. 1–29. INRIA, France (1994)
7. Meer, P., Georgescu, B.: Edge detection with embedded confidence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-23, 1351–1365 (2001)
8. Roberts, L.G.: *Machine perception of 3-D solids*. Optical and Electro-Optical Information Processing. MIT Press (1965)
9. Forsyth, D.A., Ponce, J.: *Computer Vision: “A Modern Approach”*. Prentice Hall, Upper Saddle River (2003)
10. Marr, D., Hildreth, E.: Theory of Edge Detection. In: *Proceedings of the Royal Society of London*, February 29. Series B, Biological Sciences, vol. 207(1167), pp. 187–217 (1980)
11. Smith, S.M., Brady, J.M.: SUSAN: A new approach to low level image processing. *International Journal of Computer Vision* 23(1), 45–78 (1987)
12. Zheng, Z., Wang, H., Teoh, E.K.: Analysis of Gray Level Corner Detection. *Pattern Recognition Letters* 20(2), 149–162 (1999)
13. Verma, O.P., Hanmandlu, M., Kumar, P., Chhabra, S., Jindal, A.: A novel bacterial foraging technique for edge detection. *Pattern Recognition Letters* 32(8), 1187–1196 (2011)
14. Shannon, C.E.: A mathematical theory of communication. *Bell System Technical Journal* 27, 379–423 (1948)
15. Al Sharhan, S.: Fuzzy entropy: A brief survey. In: *The 10th IEEE International Conference on Fuzzy Systems*, pp. 1135–1139 (2001)
16. Bloch, I.: Fuzzy sets in image processing. In: *Proceedings of ACM Symposium on Applied Computing*, New York, USA, March 6-8, pp. 175–179 (1994)
17. Bezdek, J.C., Chandrasekhar, R., Attikiouzel, Y.: A geometric approach to edge detection. *IEEE Transactions on Fuzzy Systems* 6(1), 52–75 (1998)
18. Boskovitz, V., Guterman, H.: An adaptive neuro-fuzzy system for automatic Image segmentation and edge detection. *IEEE Transactions on Fuzzy Systems* 10(2), 247–261 (2002)
19. Becerikli, Y., Karan, T.M.: A new fuzzy approach for edge detection. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) *IWANN 2005*. LNCS, vol. 3512, pp. 943–951. Springer, Heidelberg (2005)
20. Aborisade, D.O.: Fuzzy Logic Based Digital Image Edge Detection. *Global Journal of Computer Science and Technology* 10(14), (Ver. 1.0), 78–83 (2010)
21. Barrenechea, E., Bustince, H., De Baets, B., Lopez-Molina, C.: Construction of interval-valued fuzzy relations with application to the generation of fuzzy edge images. *IEEE Transactions on Fuzzy Systems* 19(5), 819–830 (2011)
22. Russo, F., Ramponi, G.: Edge extraction by FIRE operators. In: *Proc. of 3rd IEEE Conference on Fuzzy Systems*, pp. 249–253 (June 1994)
23. Russo, F.: FIRE operators for image processing. *Fuzzy Sets and Systems* 103(2), 265–275 (1999)
24. Hoepfner, F.: Fuzzy Shell Clustering Algorithms in Image Processing: “Fuzzy C-Rectangular and 2-Rectangular Shells”. *IEEE Trans. on Fuzzy Systems* 5(4), 599–613 (1997)

25. 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)
26. Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computing* 15(1) (February 2011)
27. Zhang, J., Sanderson, A.C.: JADE: Adaptive Differential Evolution with Optional External Archive. *IEEE Transactions on Evolutionary Computing* (2008)
28. Otsu, N.: A threshold selection method from gray level histogram. *IEEE Trans. Syst., Man, Cybern.* 9, 62–66 (1979)
29. Chow, C.K., Kaneko, T.: Automatic boundary detection of the left ventricle from cineangiograms. *Computers and Biomedical Research* 5(4), 388–410 (1972)
30. Bernsen, J.: Dynamic thresholding of gray level images. In: *Proceeding of IEEE International Conference on Pattern Recognition*, pp. 1251–1255 (1986)
31. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: *Proceedings of IEEE International Conference on Neural Network*, Perth, Australia, pp. 1942–1948. IEEE Service Center, Piscataway (1998)
32. Cohen, J.: A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20, 37–46 (1960)
33. Wesolkowski, S.B.: Color Image Edge Detection and Segmentation: A Comparison of the Vector Angle and the Euclidean Distance Color Similarity Measures. A thesis presented to the University of Waterloo in fulfilment of the thesis requirement for the degree of Master of Applied Science in Systems Design Engineering Waterloo, Ontario, Canada (1999)

A Differential Evolution Approach to Multi-level Image Thresholding Using Type II Fuzzy Sets

Ritambhar Burman¹, Sujoy Paul¹, and Swagatam Das²

¹ Department of Electronics and Tele-Communication Engineering,
Jadavpur University, Kolkata-700 078, India

² Electronics and Communication Sciences Unit, India Statistical Institute,
Kolkata-700 108, India

{sujoy_pacific, swagatamdas19}@yahoo.co.in,
ritambharkcdburman@gmail.com

Abstract. Multi-level image thresholding is an important aspect in many image processing and computer vision applications. In the last decade, many fuzzy based image thresholding techniques have been proposed. In this article a new method for multi-level image thresholding is proposed using Type II Fuzzy sets. A new entropy measure is defined which is maximized to obtain the optimal thresholds for an image. As the number of thresholds increases, exhaustive search appears to be very time consuming. So, Differential Evolution (DE), a meta-heuristic algorithm, is used for fast selection of optimal thresholds. The proposed algorithm is compared with a fuzzy entropy based algorithm using image quality assessment measures Feature Similarity Index Measurement (FSIM) and Gradient Similarity Measurement (GSM). The use of DE is also justified by comparing it with other modern state-of-art algorithms like Gravitational Search Algorithm (GSA), Particle Swarm Optimization (PSO) and Genetic Algorithm (GA).

1 Introduction

Multi-Level Image thresholding is a method in which an image is segmented into various objects. Over the years it is being applied as a basic step for several computer vision applications like pattern recognition, feature extraction, identification, image registration etc. *Otsu* [1] developed a non-parametric multilevel image segmentation algorithm. Many entropy based algorithm has also been proposed like Shannon entropy [2], Tsalli's entropy [3], Renyi's entropy [4-5] and a fuzzy entropy based algorithm [6]. But the demerits, like longer computational time and complexity, could not be avoided. Zhao et al. [7] thresholded the image by partitioning the histogram using fuzzy membership values and derived a condition for optimal threshold selection, which was later modified by Tao et al.[8] where they proposed a fuzzy entropy based technique. They partitioned the image histogram into various objects using fuzzy partition.

Type II Fuzzy sets is a generalization of Type I fuzzy sets. A measure called ultra fuzziness is associated with image segmentation using Type II Fuzzy Sets, which has

been used to obtain the optimal image thresholds in [9-12]. Although many algorithms for bi-level image segmentation using Type II Fuzzy Sets have been proposed, it has not been explored for multi-level segmentation. In this paper, a new Type II Fuzzy based multi-level image segmentation algorithm is proposed. A measure Type II Fuzzy entropy is introduced, which is maximized in order to obtain the best thresholds. But as the number of thresholds increases, the time required for execution will increase almost exponentially. For this reason a powerful meta-heuristic Differential Evolution (DE) is used for fast convergence and less computational time. Although there exists many other meta-heuristic like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Gravitational Search Algorithm(GSA), Ant Colony Optimization (ACO), Stimulated Annealing (SA), Bacteria Forging Optimization (BFO), etc, DE stands out to best for optimizing the proposed fitness function, which is discussed in section 4.

The rest of the paper is organized as follows: Section 2 provides a very brief introduction to Type II Fuzzy Sets and the proposed algorithm of multi-level image segmentation. Section 3 provides a brief introduction to Differential Evolution and section 4 shows the experimental results and comparisons. Lastly we conclude in section 5. The proposed algorithm is compared with [8], which is a fuzzy type I entropy based algorithm, which has been referred as Fuzzy Type I for the rest of the paper.

2 Multilevel Thresholding Using Fuzzy Type II Sets

A fuzzy set A [13-14], in a finite set $X = \{x_1, x_2, \dots, x_n\}$ may be represented as

$$A = \{x, \mu_A(x) | x \in X, 0 \leq \mu_A(x) \leq 1\}$$

These sets are generally called Type-I fuzzy sets. Although membership values are assigned to the elements of this set, it cannot handle much uncertainty. For this reason Type-II fuzzy sets was introduced in which a range of membership values are used instead of a single value. It may be defined as:

$$A = \left\{ x, \overline{\mu}_A(x), \underline{\mu}_A(x), \mid x \in X, 0 \leq \overline{\mu}_A(x), \underline{\mu}_A(x) \leq 1 \right\} \tag{1}$$

where $\overline{\mu}_A(x)$ and $\underline{\mu}_A(x)$ are the upper and lower membership functions respectively. Let a digital image I be of the size of $M * N$. Let $f(m, n)$ be the gray value of the pixel (m, n) where $m \in \{1, 2, \dots, M\}$ and $n \in \{1, 2, \dots, N\}$. If L is the number of gray levels by which image I can be represented, the set of all gray levels $\{0, 1, 2 \dots, L - 1\}$ is represented as G :

$$D_k = \{(m, n): f(m, n) = k, k \in \{0, 1, 2 \dots, L - 1\}\} \tag{2}$$

Let $H = \{h_0, h_1, \dots, h_{L-1}\}$ be the normalized histogram of the image where $h_k = n_k / (M * N)$, n_k is the number of pixels in D_k . A measure ultra-fuzziness is associated with a fuzzy set. If the membership values can be defined without any uncertainty then ultra-fuzziness should be 0, whereas when membership values can

only be indicated as an interval, the amount of ultra-fuzziness will increase until it reaches a maximum value of 1. The ultra-fuzziness for an image for the k^{th} level may be defined mathematically as:

$$P_k = \sum_{i=0}^{L-1} h_i * (\overline{\mu}_k(i) - \underline{\mu}_k(i)) \tag{3}$$

where $k \in \{1, \dots, N + 1\}$, μ_k is the trapezoidal fuzzy membership function for the gray levels to belong to the k^{th} level out of $N + 1$ levels of segmentation, as shown in fig 1. $\overline{\mu}_k$ and $\underline{\mu}_k$ are the upper and lower fuzzy membership functions as described in eqn. 1. Here we have used $\overline{\mu}_k = (\mu_k)^{1/\alpha}$ and $\underline{\mu}_k = (\mu_k)^\alpha, \alpha \in (1,2]$. The value of α considered by us is 1.5. μ_k can be mathematically expressed as:

$$\mu_k(i) = \begin{cases} 0 & , i \leq a_{k-1} \\ \frac{i - a_{k-1}}{c_{k-1} - a_{k-1}} & , a_{k-1} < i \leq c_{k-1} \\ 1 & , c_{k-1} < i \leq a_k \\ \frac{i - c_k}{a_k - c_k} & , a_k < i \leq c_k \\ 0 & , i > c_k \end{cases} \tag{4}$$

where a_k and $c_k, k \in \{1,2, \dots, N + 1\}$, are the fuzzy parameters and $a_0 = c_0 = 0, a_{N+1} = c_{N+1} = L - 1, N$ being the number of thresholds. The fuzzy type II entropy for the k^{th} level of segmentation may be defined as:

$$H_k = - \sum_{i=0}^{L-1} \frac{h_i * (\overline{\mu}_k(i) - \underline{\mu}_k(i))}{P_k} * \ln \left(\frac{h_i * (\overline{\mu}_k(i) - \underline{\mu}_k(i))}{P_k} \right) \tag{5}$$

$k \in \{1,2, \dots, N + 1\}$. The total entropy is the summation of the entropies for all the levels,

$$H(a_1, c_1, \dots, a_n, c_n) = \sum_{k=1}^{N+1} H_k \tag{6}$$

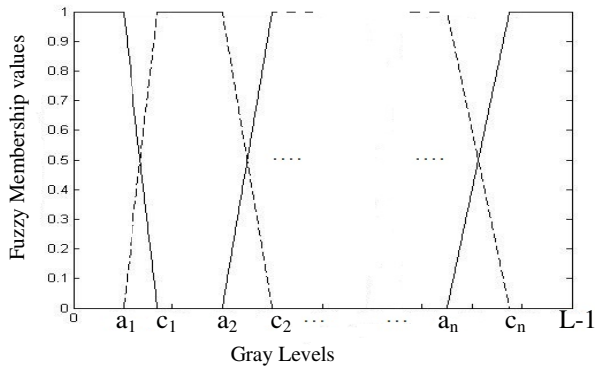
The optimal fuzzy parameters can be found out by maximizing the total entropy,

$$(a_1, c_1, \dots, a_n, c_n)^* = Argmax(H) \tag{7}$$

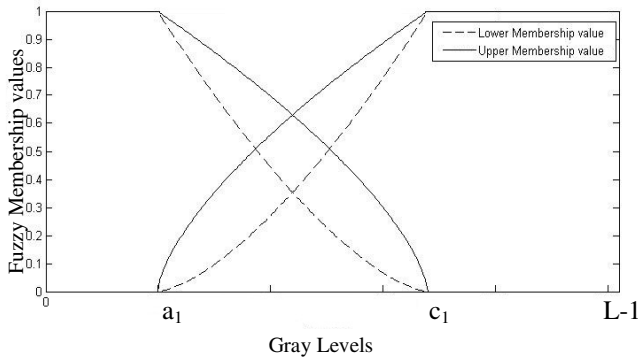
The thresholds for the image is that point, where the membership value of a level of segmentation falls to 0.5, and as the membership functions are linear, it may be written that the n^{th} threshold is,

$$T_n = \frac{1}{2}(a_n + c_n), n \in \{1,2, \dots, N\} \tag{8}$$

where N is the number of thresholds. DE is used to reduce the search time to find the optimal thresholds. The parameter set $(a_1, c_1, \dots, a_n, c_n)$ acts as an individual of the population in DE.



1.1



1.2

Fig. 1. 1.1: Fuzzy membership function for n+1 level of segmentation, 1.2: An example of upper and lower approximation membership function for bi-level segmentation

Recursive method for Type II Fuzzy Entropy calculation

```

Type2entropy(v) // v=[v1,v2,v3,...vn] is the vector of fuzzy parameters
if (n==4) //n is the total number elements in v
    μ ← trapezoidal(v) //trapezoidal() returns the fuzzy membership values using eqn. 5
    μ̄ ← μ1/α, μ̄ = μα //Both the operations are done element wise
    P ← ∑i=0L-1 (μ̄(i) - μ(i)) * h(i) //h is the normalized histogram of the image
    H ← - ∑i=0L-1  $\frac{h(i) * (\bar{\mu}(i) - \underline{\mu}(i))}{P} * \ln \left( \frac{h(i) * (\bar{\mu}(i) - \underline{\mu}(i))}{P} \right)$ 
else
    H ← Type2entropy([v1,v2,v3,v4]) + Type2entropy([v5,v6,...,vn])
end if
end of function

```

Two dummy threshold values, 0 and L-1 are added with the fuzzy parameters, when calling this function. So the format of function call will be *Type2entropy* ($[0, 0, a_1, c_1, a_2, c_2 \dots, a_n, c_n, L-1, L-1]$)

3 Differential Evolution (DE)

DE is a global optimization technique, which was proposed by Storn [15]. DE searches for a global optimum point in a D -dimensional real parameter space R^D . In our algorithm, a simple version of DE, named DE/rand/1 scheme is used. The i^{th} individual of the population which is a vector of dimension D can be represented as,

$$\vec{X}_i(t) = [\vec{X}_{i,1}(t), \vec{X}_{i,2}(t), \dots, \vec{X}_{i,D}(t)]$$

The first step is initialization, where the population is initialized randomly within the search space. This is done as shown below,

$$\vec{X}_{i,j}(t) = X_{jmin} + rand * (X_{jmax} - X_{jmin}), i = 1 \text{ to } NP \text{ and } j = 1 \text{ to } D \quad (9)$$

where X_{jmax} and X_{jmin} are the maximum and minimum bound of the search space, NP is the total population involved in the search process and $rand$ is a random number in between 0 and 1. In every iteration, for each parent a vector, a mutant vector, called donor vector, is obtained through the differential mutation operation. To create the i^{th} donor vector $\vec{Y}_i(t)$ for the i^{th} parent vector, three other parent vectors (say the $a1, a2$ and $a3$ -th vectors such that $a1, a2, a3 \in [1, NP]$ and $a1 \neq a2 \neq a3$) are chosen at random from the current population. The donor vector may be expressed as,

$$\vec{Y}_i(t) = \vec{X}_{a1}(t) + F * (\vec{X}_{a2}(t) - \vec{X}_{a3}(t)) \quad (10)$$

where F is a scalar quantity called Weighting Factor. The next step is to enhance the potential diversity of the donor vectors to create the trial vectors. A binomial crossover operation is performed on each of the D variables of a vector as,

$$\begin{aligned} R_{i,j}(t) &= Y_{i,j}(t), \text{ if } rand_j(0,1) \leq Cr \text{ or } j = j_{rand} \\ &= X_{i,j}(t), \text{ otherwise} \end{aligned} \quad (11)$$

where $j = 1 \text{ to } D$ and $rand_j \in [0,1]$ is the j^{th} evaluation of a uniform random generator. $j_{rand} \in [1, 2, \dots, D]$, is a randomly chosen index to ensure that $\vec{R}_i(t)$ gets at least one component of $\vec{X}_i(t)$ and Cr is the crossover rate. The trial vector is updated as the new parent vector for iteration $t+1$, if its fitness function value is more (for maximization) than current parent vector, in iteration t .

$$\begin{aligned} \vec{X}_i(t+1) &= \vec{R}_i(t), \text{ if } f(\vec{R}_i(t)) \geq f(\vec{X}_i(t)), \\ &= \vec{X}_i(t), \text{ if } f(\vec{R}_i(t)) \leq f(\vec{X}_i(t)) \end{aligned}$$

The above steps are repeated until the termination criterion is reached. We have used number of iterations as the termination criterion. An extensive survey on DE may be found in [16].

4 Experimental Results

The simulations are performed in Matlab 2012a in a workstation with Intel Core i3 2.9 GHz processor. For testing and comparison, 4 images have been chosen arbitrarily from Berkeley Segmentation Dataset and Benchmark [17].

Table 1. Set-up for DE

| | |
|--------------------------------------|---|
| No. of Runs | 20 |
| No. of iterations per run | 100 |
| Dimension of the search space(D) | $2 * \text{No. of Thresholds}$ |
| Upper bound of search space | Maximum gray level present in the image |
| Lower bound of search space | Minimum gray level present in the image |
| Number of Particles(NP) | $10 * D$ |
| Weighing Factor(F) | 0.5 |
| Crossover Rate(Cr) | 0.9 |

We have compared the proposed Type II Fuzzy method with Type I Fuzzy entropy based algorithm as in [8]. The test images thresholded by the proposed algorithm and the Type I Fuzzy method is shown in fig 2 for 2, 3, 4 thresholds. The image number corresponds to the number provided in the dataset. The fuzzy membership function parameters and the corresponding thresholds for the test images is listed in table 2 and in table 3 for Fuzzy Type II proposed method and Type I method.

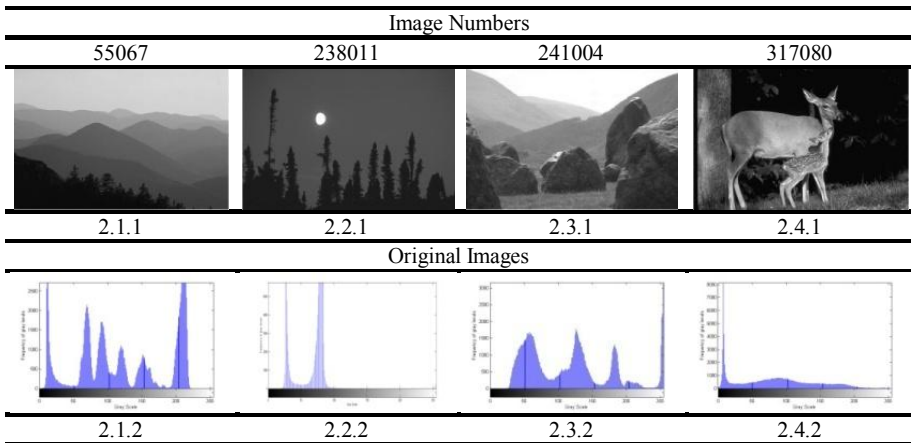


Fig. 2. Test images thresholded by Fuzzy Type I and II method

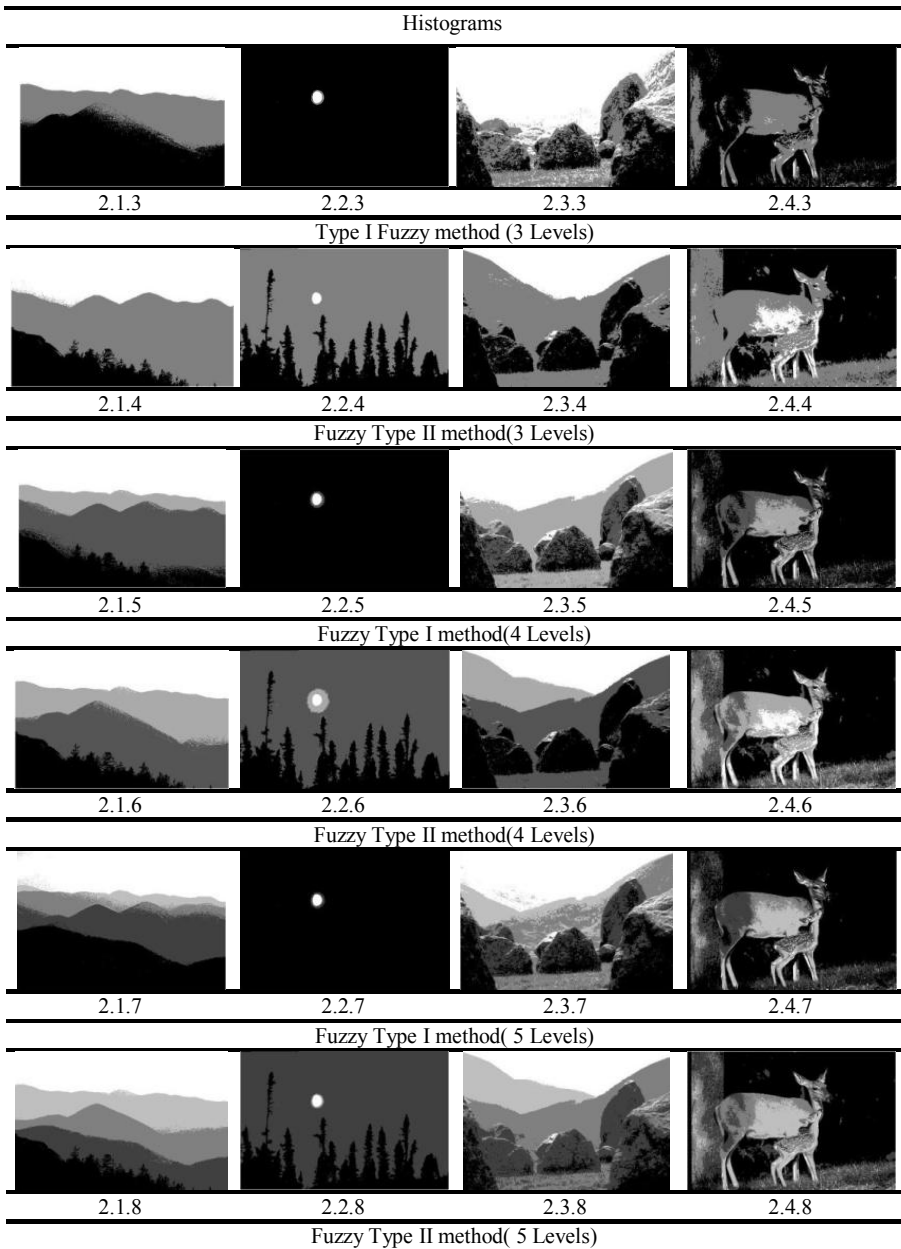


Fig. 2. (continued)

Table 2. Fuzzy parameters and thresholds for images thresholded by Fuzzy Type II method

| Image No. | No. of thresholds (T _N) | Fuzzy Parameters | Thresholds |
|-----------|-------------------------------------|-------------------------------|----------------|
| 238011 | 2 | 28,73,110,255 | 51,183 |
| | 3 | 28,74,94,94,107,252 | 51,94,180 |
| | 4 | 29,72,86,113,113,113,113,252 | 51,100,113,183 |
| 241004 | 2 | 25,131,131,255 | 78,193 |
| | 3 | 25,142,142,185,185,254 | 84,164,220 |
| | 4 | 25,75,75,136,136,186,186,254 | 50,106,161,220 |
| 317080 | 2 | 8,96,96,252 | 52,174 |
| | 3 | 8,105,105,105,105,252 | 57,105,179 |
| | 4 | 8,102,102,102,102,203,203,252 | 55,102,153,228 |
| 55067 | 2 | 12,63,63,219 | 38,141 |
| | 3 | 12,64,64,133,133,217 | 38,99,175 |
| | 4 | 12,66,66,92,92,131,131,215 | 39,79,112,173 |

In fig 2.1.3, it may be seen that the bushes merged with the mountains is not visible, but it is properly segmented by the proposed algorithm in fig 2.1.4. Also in fig 2.1.3, the transition in mountains are not clearly defined, due to improper choice of thresholds, which is properly done by the proposed algorithm, which segments the sky, mountains and the bushes perfectly. In fig 2.2.3, only the moon is separated from the sky, but the trees are merged with the sky, which is properly segmented in fig 2.2.4. In fig 2.3.3, which segmented by Fuzzy Type I method, the mountain is merged with the sky, which is properly segmented by the proposed Fuzzy Type II method. Also the rocks are not properly segmented from the surrounding in fig 2.3.3, which is properly done in fig 2.3.4. It can be seen from the original image fig 2.3.1, there exists two mountain ranges with different gray levels, which is properly segmented from each other, when the number of thresholds is increased, as is done by the proposed algorithm in fig 2.3.8, but not in fig 2.3.7 which is segmented by Type I Fuzzy Entropy method. In fig 2.4.3, the deer is not properly segmented from the background, which is done by the proposed algorithm in fig 2.4.4. In 2.1.7, the mountain ranges are not segmented properly from each other, which is done by the proposed Type II method.

Table 3. Fuzzy parameters and thresholds for images thresholded by Fuzzy Type I method

| Image No. | No. of thresholds (T _N) | Fuzzy Parameters | Thresholds |
|-----------|-------------------------------------|--------------------------------|----------------|
| 238011 | 2 | 62,122,123,178 | 92,151 |
| | 3 | 86,106,107,159,160,255 | 96,133,208 |
| | 4 | 85,107,108,147,148,228,228,253 | 96,128,188,241 |
| 241004 | 2 | 25,97,98,148 | 61,123 |
| | 3 | 25,93,94,95,96,207 | 59,95,152 |
| | 4 | 25,94,95,98,99,164,164,190 | 60,97,132,177 |

Table 4. (continued)

| | | | |
|--------|---|--------------------------------|----------------|
| 317080 | 2 | 62,174,175,252 | 118,214 |
| | 3 | 64,131,132,181,182,252 | 98,57,217 |
| | 4 | 23,122,123,130,131,191,191,252 | 73,127,161,222 |
| 55067 | 2 | 55,134,135,222 | 95,179 |
| | 3 | 6,127,128,129,130,222 | 67,129,176 |
| | 4 | 52,112,113,134,135,174,174,217 | 82,124,155,196 |

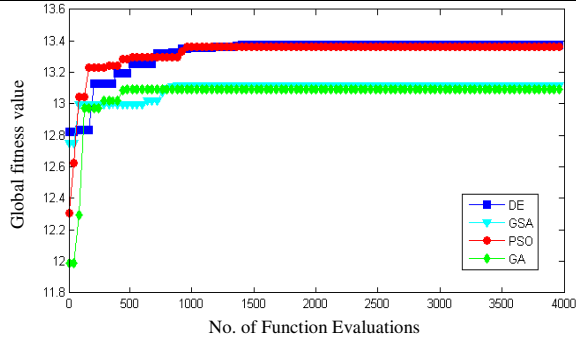
Table 5. FSIM and GSM of test images

| Image No. | No. Of thresholds (T_N) | Fuzzy Type I | | Fuzzy Type II | |
|-----------|-----------------------------|--------------|--------|---------------|---------------|
| | | FSIM | GSM | FSIM | GSM |
| 238011 | 2 | 0.6624 | 0.9749 | 0.8379 | 0.9797 |
| | 3 | 0.6582 | 0.9746 | 0.8594 | 0.9882 |
| | 4 | 0.6598 | 0.9747 | 0.8702 | 0.9908 |
| 241004 | 2 | 0.5908 | 0.9261 | 0.6794 | 0.9599 |
| | 3 | 0.6931 | 0.9557 | 0.7177 | 0.9694 |
| 317080 | 4 | 0.7163 | 0.9595 | 0.7479 | 0.9699 |
| | 2 | 0.6153 | 0.9512 | 0.6372 | 0.9527 |
| | 3 | 0.6453 | 0.9605 | 0.6893 | 0.9645 |
| 55067 | 4 | 0.7141 | 0.9705 | 0.7481 | 0.9751 |
| | 2 | 0.7661 | 0.9697 | 0.8058 | 0.9782 |
| | 3 | 0.8061 | 0.9766 | 0.8268 | 0.9811 |
| | 4 | 0.7781 | 0.9749 | 0.8288 | 0.9827 |

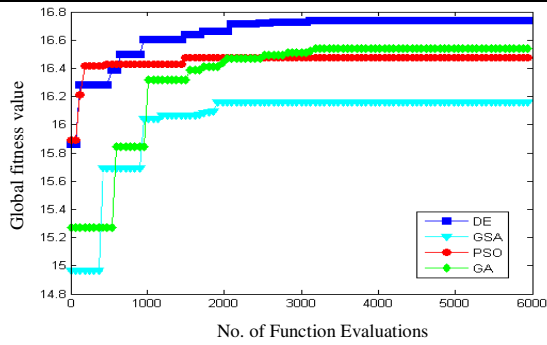
Table 6. Mean fitness function value and its standard deviation

| Image No. | T_N | GSA | | PSO | | GA | | DE | |
|-----------|-------|------------|-------------|------------|-------------|------------|-------------|----------------|-----------------|
| | | f_{mean} | <i>Std.</i> | f_{mean} | <i>Std.</i> | f_{mean} | <i>Std.</i> | f_{mean} | <i>Std.</i> |
| 38011 | 2 | 11.4316 | 1.21E-01 | 11.3577 | 2.36E-02 | 11.3568 | 2.07E-01 | 11.4907 | 2.36E-07 |
| | 3 | 14.9668 | 1.01E-01 | 15.0499 | 2.31E-01 | 15.0733 | 1.77E-01 | 15.5711 | 9.11E-15 |
| | 4 | 17.9144 | 2.79E-01 | 17.9814 | 1.21E-01 | 18.1618 | 1.84E-01 | 18.5434 | 3.15E-06 |
| 241004 | 2 | 13.5425 | 4.58E-02 | 13.5652 | 2.28E-02 | 13.5568 | 7.06E-02 | 13.6097 | 6.69E-06 |
| | 3 | 17.1457 | 2.86E-01 | 17.3599 | 1.12E-01 | 17.2973 | 2.14E-01 | 17.5498 | 3.65E-15 |
| 317080 | 4 | 20.7938 | 1.18E-01 | 20.4551 | 1.31E-01 | 20.8065 | 4.39E-02 | 20.8677 | 1.09E-14 |
| | 2 | 14.1098 | 7.02E-02 | 14.2623 | 3.22E-02 | 14.2431 | 9.29E-02 | 14.3238 | 7.87E-12 |
| | 3 | 17.9488 | 8.21E-02 | 18.0022 | 1.65E-01 | 18.1431 | 5.76E-02 | 18.2541 | 8.29E-15 |
| 55067 | 4 | 21.4309 | 9.61E-02 | 21.2355 | 1.08E-01 | 21.4835 | 6.47E-02 | 21.6667 | 3.15E-08 |
| | 2 | 13.1160 | 8.17E-02 | 13.3602 | 1.14E-01 | 13.0923 | 1.04E-01 | 13.3739 | 2.01E-09 |
| | 3 | 16.1581 | 1.71E-01 | 16.4728 | 1.24E-01 | 16.5837 | 1.30E-01 | 16.7403 | 5.40E-06 |
| | 4 | 19.4894 | 1.79E-01 | 19.7575 | 2.04E-01 | 19.4033 | 1.20E-01 | 19.9396 | 2.67E-09 |

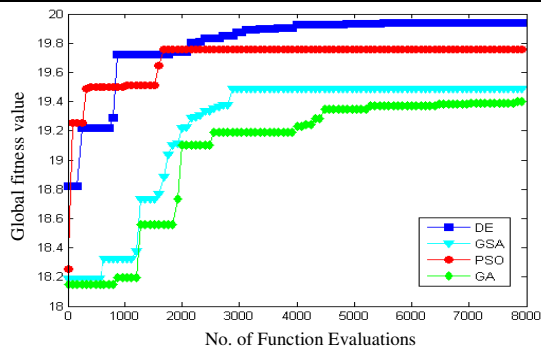
Two modern Image Quality Assessment (IQA) measures, namely Feature Similarity Index (FSIM) [18] and Gradient Similarity Measurement (GSM) [19] are used for quantitative comparisons. The measures for the thresholded images in fig 2 are listed in table 3(Higher the value of the measure, better is the segmentation). It may be noticed that the proposed algorithm dominates the Fuzzy Type I method of image segmentation.



3.1



3.2



3.3

Fig. 3. Convergence plots. 3.1, 3.2, 3.3 are for 2, 3, 4 thresholds respectively

Performance of DE is compared with some modern state-of-art algorithms like Particle Swarm Optimization (PSO) [20], Genetic Algorithm (GA) [21] and a relatively recent optimizer, Gravitational Search Algorithm (GSA) [22]. Although many variants of each optimizer exist, the basic algorithm of all the optimizers is considered here for comparison. In table 4 and 5, the mean fitness value, their standard deviation calculated over 20 runs, with 100 iterations in each run and

Table 7. Average computational time consumed by DE, GSA, PSO and GA

| Image No. | No. Of thresholds (T_N) | Computational time (in second) | | | |
|-----------|-----------------------------|--------------------------------|--------|---------|---------------|
| | | GSA | PSO | GA | DE |
| 238011 | 2 | 1.8586 | 1.6535 | 5.9674 | 1.6031 |
| | 3 | 3.2792 | 3.0401 | 10.6179 | 2.9307 |
| | 4 | 5.1945 | 4.5844 | 17.2011 | 4.5592 |
| 241004 | 2 | 1.9042 | 1.8075 | 6.3232 | 1.7557 |
| | 3 | 3.5769 | 3.2537 | 11.2242 | 3.1678 |
| | 4 | 5.5361 | 4.8719 | 17.8267 | 4.8354 |
| 317080 | 2 | 1.9974 | 1.8558 | 6.3868 | 1.8154 |
| | 3 | 3.5332 | 3.2297 | 11.4714 | 3.1114 |
| | 4 | 5.6833 | 5.0335 | 17.6465 | 4.9289 |
| 55067 | 2 | 1.8709 | 1.7624 | 6.1427 | 1.6993 |
| | 3 | 3.5264 | 3.1718 | 11.1587 | 3.0829 |
| | 4 | 5.5268 | 4.8462 | 17.2196 | 4.7554 |

computational time is listed. It may be seen that DE requires least computational time, followed by PSO which takes almost same time as DE, but on the higher side, followed by GSA and maximum by GA, which takes almost 4-5 times the time consumed by DE. It may be noticed that the optimum fitness value for all the images are maximum for DE with respect to the other optimizers. Also the standard deviation of the optimum fitness value is minimum for all the images when optimized by DE. The convergence plots for DE, PSO, GSA and GA are shown for image 55067 in fig 3.1, 3.2, 3.3 for 2, 3, 4 thresholds respectively. It may be seen from the convergence plots that when the number of thresholds are less, DE and PSO have almost same convergence rate. As the number of thresholds increases, the difference between the convergence plots increases. But in all the cases, DE has the maximum optimized fitness value.

5 Conclusion

An algorithm of image thresholding should be such that it segments the image into different objects accurately and at the same time consumes as less time as possible to accomplish the task. The proposed algorithm based on Type II Fuzzy sets produces better results than the Type I Fuzzy entropy based algorithm. The use of DE speeds up the search process. Results reflect that DE dominates meta-heuristics. Future work may be undertaken to improve this algorithm to segment the images more accurately and consume lesser time, as required for very fast real time image segmentation and patten recognition problems.

References

1. Otsu, N.: A threshold selection method from gray level histograms. IEEE Transactions on System, Man and Cybernetics 9, 62–66 (1979)
2. Benzid, R., Arar, D., Bentoumi, M.: A fast technique for gray level image thresholding and quantization based on the entropy maximization. In: 5th International Multi-Conference on Systems, Signals and Devices, pp. 1–4 (2008)

3. Sarkar, S., Das, S., Chaudhuri, S.S.: Multilevel Image Thresholding Based on Tsallis Entropy and Differential Evolution. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Nanda, P.K. (eds.) SEMCCO 2012. LNCS, vol. 7677, pp. 17–24. Springer, Heidelberg (2012)
4. Sahoo, P.K., Arora, G.: A thresholding method based on two dimensional Renyi's entropy. *Pattern Recognition* 37, 1149–1161 (2004)
5. Lan, S., Kong, L.L.Z., Wang, J.G.: Segmentation Approach Based on Fuzzy Renyi Entropy. *Pattern Recognition*, 1–4 (2010)
6. Tian, W.J., Geng, Y., Liu, J.C., Ai, L.: Maximum Fuzzy Entropy and Immune Clone Selection Algorithm for Image Segmentation. In: *Information Processing, Asia-Pacific Conference*, vol. 1, pp. 38–41 (2009)
7. Zhao, M.S., Fu, A.M.N., Yan, H.: A technique of three level thresholding based on probability partition and fuzzy 3-partition. *IEEE Trans. Fuzzy Systems* 9(3), 469–479 (2001)
8. Tao, W.B., Tian, J.W., Liu, J.: Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm. *Pattern Recognition Letters* 24, 3069–3078 (2003)
9. Tizhoosh, H.R.: Image thresholding using type II fuzzy sets. *Pattern Recognition* 38, 2363–2372 (2005)
10. Arifin, A.Z., Heddyanna, A.F., Studiawan, H.: Image thresholding using ultrafuzziness optimization based on type II fuzzy sets. *Instrumentation, Communications, Information Technology, and Biomedical Engineering*, 1–6 (November 2009)
11. Wu, J., Pian, Z., Guo, L., Wnag, K., Gao, L.: Medical Image Thresholding Algorithm based on Fuzzy Set Theory. In: *2nd IEEE Conference on Industrial Electronics and Applications*, pp. 919–924 (2007)
12. Rajesh, R., Senthilkumar, N., Satheeshkumar, J.: On the Type-1 and Type-2 Fuzziness Measures for Thresholding MRI Brain Images. *FUZZ IEEE*, 992–995 (2011)
13. Zadeh, L.A.: *Fuzzy Sets* (1965)
14. Wang, X.-J., Hao, Y.-W., Zhao, R.-H.: Method, Model and Application for the Conversion from Vague Sets to Fuzzy Sets. *Artificial Intelligence and Computational Intelligence* 1, 510–513 (2009)
15. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, 341–359 (1997)
16. Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-Art. *Evolutionary Computation* 15(1), 4–31 (2011)
17. <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench>
18. Zhang, L., Zhang, L., Mou, X., Zhang, D.: FSIM, A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing* 20(8), 2378–2386 (2011)
19. Zhu, J., Wang, N.: Image Quality Assessment by Visual Gradient Similarity. *IEEE Transactions on Image Processing* 21(3), 919–933 (2012)
20. Kennedy, J., Eberhat, R.: Particle swarm optimization. *IEEE International Conference on Neural Networks* 4, 1942–1948 (1995)
21. Deb, K., Anand, A., Joshi, D.: A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation* 10(4), 371–395 (2002)
22. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: A Gravitational Search Algorithm 179(13), 2232–2248 (2009)

Differential Evolution with Controlled Annihilation and Regeneration of Individuals and A Novel Mutation Scheme

Sudipto Mukherjee¹, Sarthak Chatterjee¹,
Debdipta Goswami¹, and Swagatam Das²

¹ Department of Electronics and Telecommunication Engineering
Jadavpur University, Kolkata, India

² Electronics and Communication Sciences Unit
Indian Statistical Institute, Kolkata, India

Abstract. Differential Evolution is a stochastic, population-based optimization algorithm, which grew out of the need to optimize real-parameter, real-valued functions. The Differential Evolution variant that we propose to describe in this paper modifies the mutation scheme of the variant DE/best/1. We propose a three tier mutation scheme, to be suitably carried out on selected sections of the population in question. Also, the proposed variant tries to lessen the myriad troubles posed by stagnation, which is a problem faced by all Differential Evolution algorithms. Our comparative studies indicate that the proposed variant is able to compete in a direction parallel to the state-of-the-art Differential Evolution variants like JADE and jDE.

Keywords: optimization, Differential Evolution(DE), novel mutation scheme, stagnation.

1 Introduction

Differential Evolution [6], [8]-[10] emerged in the late 1990s to serve the immediate need of mathematicians, engineers and technicians to solve real-world optimization problems. Over the years, cumulative research on differential evolution and its sundry varieties has reached an impressive state. Modifications have been proposed by introducing innovative mutation schemes, schemes that better implement crossover, and tuning parameters like the scale factor, F and the crossover ratio, Cr .

In this paper, we propose two new algorithmic components, which can be used to improve results. They are given as follows:

1. The mutation scheme of DE/best/1 has been retained but a provision has been made to change the mutation schemes for three different sections of the population.
2. A sincere effort has been made to remove the problem of stagnation by introducing the control parameter *stagnate* when there is no improvement of

global best fitness value after mutation, crossover and selection for certain successive generations. If this happens, we annihilate and regenerate the stagnation-causing portion of the population.

The proposed DE variant is compared with DE/rand/1/bin, DE/current-to-best/1/bin, SaDE, JADE, jDE and DEGL over 25 standard numerical benchmark functions taken from the CEC 2005 competition and special session on real-parameter optimization.

The paper is organized as follows: Section II gives a general overview of the DE family of algorithms. Section III explains the essential features of the proposed DE variant. The experimental settings for the benchmarks and simulation strategies are presented in section IV along with results that outline the performance of the algorithm. Finally section V, presents a short discussion about the probable applications of the algorithm in prospective areas.

2 Differential Evolution : A General Discussion

Differential Evolution is one method which is a member of a class of methods called metaheuristics. It is an iterative scheme developed by Storn and Price in 1997 [8] which seeks to minimize an objective function by iteratively seeking a parameter vector \mathbf{X}^* which minimizes the objective function $f(\mathbf{X}^*)$ ($f : \Omega \subseteq \mathbb{R}^D \rightarrow \mathbb{R}$), that is $f(\mathbf{X}^*) < f(\mathbf{X})$ where $\mathbf{X} = [x_1, x_2, x_3, \dots, x_D]^T$, the parameter vector which characterizes the performance of a system, for all $\mathbf{X} \in \Omega$, where Ω is a non-empty, large, finite set serving the purpose for the domain of the search.

2.1 Initialization of the Vectors Controlling the Performance of the System

Differential Evolution searches for a global optimum in a search space \mathbb{R}^D comprising of D dimensions. The first step is to initialize a population of NP , D dimensional, real-valued parameter vectors, where NP is the population size for the optimization problem at hand. We call each such vector a genome or chromosome. In analogy with biological processes, the genomes will modify their values over generations which may be denoted by $G = 0, 1, 2, \dots, G_{max}$. The i th vector of the population for the current generation may be represented by

$$\mathbf{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \dots, x_{D,i,G}].$$

The initialization of the population at $G = 0$ must be done taking care of the bounds which restrict the parameter vector \mathbf{X} . As such, the minimum and maximum values for \mathbf{X} are given by

$$\mathbf{X}_{min} = [x_{1,min}, x_{2,min}, x_{3,min}, \dots, x_{D,min}],$$

$$\mathbf{X}_{max} = [x_{1,max}, x_{2,max}, x_{3,max}, \dots, x_{D,max}].$$

The initialization of the j th component of the i th vector is done as follows

$$x_{j,i,0} = x_{j,min} + rand_{i,j}[0, 1] (x_{j,max} - x_{j,min}),$$

where $rand_{i,j}[0, 1]$ is a uniformly distributed random number lying between 0 and 1 ($0 \leq rand_{i,j}[0, 1] \leq 1$) and is instantiated independently for each component of the i th vector. This is done so as to ensure that the population initialized at $G = 0$ covers, as far as possible, the range of values that can possibly be taken by \mathbf{X} .

2.2 Mutation with Difference Vectors

Once the initialization has been done, the objective of the algorithm is to create a donor vector $\mathbf{V}_{i,G}$ corresponding to each member of the population (rechristened as the target vector, $\mathbf{X}_{i,G}$) in the current generation through a process called mutation. The five most commonly used mutation schemes are as follows:

1. DE/rand/1

$$\mathbf{V}_{i,G} = \mathbf{X}_{r_1^i,G} + F(\mathbf{X}_{r_2^i,G} - \mathbf{X}_{r_3^i,G})$$

2. DE/best/1

$$\mathbf{V}_{i,G} = \mathbf{X}_{best,G} + F(\mathbf{X}_{r_1^i,G} - \mathbf{X}_{r_2^i,G})$$

3. DE/current-to-best/1

$$\mathbf{V}_{i,G} = \mathbf{X}_{i,G} + F(\mathbf{X}_{best,G} - \mathbf{X}_{i,G}) + F(\mathbf{X}_{r_1^i,G} - \mathbf{X}_{r_2^i,G})$$

4. DE/best/2

$$\mathbf{V}_{i,G} = \mathbf{X}_{i,G} + F(\mathbf{X}_{r_1^i,G} - \mathbf{X}_{r_2^i,G}) + F(\mathbf{X}_{r_3^i,G} - \mathbf{X}_{r_4^i,G})$$

5. DE/rand/2

$$\mathbf{V}_{i,G} = \mathbf{X}_{r_1^i,G} + F(\mathbf{X}_{r_2^i,G} - \mathbf{X}_{r_3^i,G}) + F(\mathbf{X}_{r_4^i,G} - \mathbf{X}_{r_5^i,G})$$

The indices r_1^i , r_2^i , r_3^i , r_4^i and r_5^i are mutually exclusive random integers chosen from the closed interval $[1, NP]$ and all of them should be different from i . These indices are randomly generated anew for each donor vector. The scale factor F is a positive control parameter for scaling the difference vectors. $\mathbf{X}_{best,G}$ is the best individual vector with the best fitness (i.e. corresponding to a particular minimization problem, $\mathbf{X}_{best,G}$ has the lowest objective function value).

2.3 The Crossover Scheme

Crossover involves the mixing of components between the donor vector and the target vector $\mathbf{X}_{i,G}$ to form the trial vector $\mathbf{U}_{i,G} = [u_{1,i,G}, u_{2,i,G}, u_{3,i,G}, \dots, u_{D,i,G}]$. The DE family of algorithms essentially uses two kinds of crossover: exponential (or two-point-modulo) and binomial (or uniform). We discuss the latter as the proposed DE variant uses it. The binomial crossover scheme may be defined as follows:

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{rand}_{i,j}[0, 1] \leq Cr, \text{ or } j = j_{rand}, \\ x_{j,i,G}, & \text{otherwise.} \end{cases}$$

where, as explained in the previous section $\text{rand}_{i,j}[0, 1]$ is a uniformly distributed random number created afresh for each j th component of the i th parameter vector. j_{rand} is a randomly chosen integer lying in the interval $[1, D]$ which ensures that $\mathbf{U}_{i,G}$ gets at least one component from $\mathbf{V}_{i,G}$. It is instantiated once for each vector for each generation. See [4] for a DE algorithm containing novel mutation and crossover strategies.

2.4 The Selection Scheme

Selection is the process that ascertains whether the target or the trial vector survives to the next generation which is denoted by $G = G + 1$. The selection operation may be described as follows:

$$\mathbf{X}_{i,G+1} = \begin{cases} \mathbf{U}_{i,G}, & f(\mathbf{U}_{i,G}) \leq f(\mathbf{X}_{i,G}), \\ \mathbf{X}_{i,G}, & f(\mathbf{U}_{i,G}) > f(\mathbf{X}_{i,G}). \end{cases}$$

where $f(\mathbf{X})$ is the given objective function to be minimized.

3 Algorithm of the Proposed Differential Evolution Variant

In this section, we slowly begin to develop the main aspects of the proposed DE variant. We start with defining the main parameters used. The scale factor, F has been given a value of 0.8, and the crossover rate, Cr has been given the value of 0.9, the latter value having been found suitable for a wide variety of optimization problems that can be successfully tackled by Differential Evolution. Parameter-selection in DE is addressed in [5]. The algorithm has been tested on a population NP of 100 individuals with the number of dimensions D being variable and set at 30, 50 and 100 respectively.

3.1 Development of a Novel Mutation Scheme that Closely Mimics Behavior Seen in the Natural World

Mutation Type I : Mutation between the Best and the Worst. The proposed DE variant uses a mutation scheme that is a slight departure from

the mutation scheme defined earlier by DE/best/1. The scheme in question is defined by

$$\mathbf{V}_{i,G} = \mathbf{X}_{best,G} + F(\mathbf{X}_{r_1^i,G} - \mathbf{X}_{r_2^i,G})$$

where r_1^i and r_2^i are mutually exclusive random integers chosen in such a way that the maximum improvement in the present generation is given utmost importance. This is done by ensuring that r_1^i is chosen from the best $p\%$ and r_2^i from the worst $p\%$ of the population. As explained earlier, the population consists of 100 individuals, and here we have chosen $p = 20$. The novelty of this approach is clearly seen if we consider the equation defined above. A close look at the term in parenthesis suggests the formation of the vector $(\mathbf{X}_{r_1^i,G} - \mathbf{X}_{r_2^i,G})$, which indicates that this vector is nothing but one that provides the direction that is approximately the direction of maximum improvement, since r_1^i and r_2^i are chosen from the best and worst $p\%$ of the population respectively.

Mutation Type II : Mutation between Two Individuals of the Best $p\%$ of the Population Apart from the above, the proposed DE variant also considers the possibility of a mutation between two randomly selected members of the best $p\%$ of the population. The proposed advantage of this type of mutation is to overcome the overtly exploratory nature of the mutation type I and to make it a bit more exploitative. Since, in the present mutation type, the mutation is being performed taking into consideration $\mathbf{X}_{r_1^i,G}$ and $\mathbf{X}_{r_2^i,G}$, which are randomly selected vectors taken from the best $p\%$ of the population, it can be expected that the two aforementioned vectors do not differ by a great deal with respect to their fitness values. The mutation type II has the cumulative effect of directing the donor vector towards the historically best vector which has the best fitness value $\mathbf{X}_{best,G}$. Hence, we find that this type of mutation maintains the best members of the present generation and progressively directs them more and more towards $\mathbf{X}_{best,G}$. Therefore, this mutation type localizes the search region of the proposed DE variant around the region of the vector $\mathbf{X}_{best,G}$.

Mutation Type III : Average Case Mutation The average case mutation type is included to maintain diversity and also the general degree of randomness that characterizes a metaheuristic. This third mutation type seeks to mutate any two randomly selected vectors of the population excluding the best $p\%$ of the population, since this case has already been considered in the first mutation type. At this juncture, it becomes an immediate necessity to differentiate between the mutation type III and the earlier described mutation types I and II.

As compared to the exploratory nature of the mutation type I and the exploitative nature of the mutation type II, the mutation type III is neither exploratory nor exploitative. Mutating between two random individuals of the best $p\%$ of the population ensures that the search is confined to contours around $\mathbf{X}_{best,G}$, and mutating between two random individuals of the best and worst $p\%$ of the population ensures that the search space is generously explored.

The overall mutation scheme, which is a combination of the types, I, II and III is carried out in the following manner: For a number of individuals equal to

$2p\%$ of the population we carry out the mutation type I, that is, a mutation between the best and worst $p\%$ of the population. For a number equal to $p\%$ of the population, the mutation type II is chosen, namely a mutation between any two members of the best $p\%$ of the population. For the remaining percentage of the population, we carry out the mutation type III, the average case mutation type elucidated earlier.

The mutation scheme is successfully executed by taking into consideration the term *Sorted_Population* which contains the individuals of the present generation but in a sorted order. The sorting is done in such a manner that the individual with the best fitness value is placed first and the individual with the worst fitness value is placed last. In other words, *Sorted_Population* contains all the individuals of the present generation, sorted according to their fitness values, the fittest being given priority.

3.2 Crossover

The proposed DE variant uses a novel crossover scheme. This elite crossover operation incorporates in its working a greedy parent-selection strategy. For each donor vector $V_{i,G}$, a vector is randomly selected from the best $p\%$ vectors of the present population and then binomial crossover is performed between the donor vector and the randomly chosen vector in order to generate the trial vector $U_{i,G}$.

3.3 Stagnation, Its Connotations and an Attempted Removal

Stagnation refers to the trapping of the population near local extrema that causes major problems by preventing the population from progressing towards the much coveted global extrema.

The stagnation in the proposed DE variant is detected by the control parameter *stagnate*. When the fitness of the population does not improve even after mutation and crossover for several generations, there arises the need for annihilation and regeneration of the so-called “bad” part of the population. The rule that has been followed for annihilation and regeneration is as follows: Here the worst $\frac{p}{2}\%$ of the population is annihilated and the same number of chromosomes is regenerated using a regeneration rule. The parameter *stagnate* is chosen such that:

1. The stagnation is detected properly.
2. The population with regenerated chromosomes gets sufficient opportunity to overcome the stagnation by improving their fitness values.

The annihilation and regeneration rule can be presented as follows: To generate a new population, each chromosome is generated according to the following rule: For every dimension j , where $j = 1, 2, 3, \dots, n$, a random number rnd_j uniformly distributed within $(0, 1)$ is generated and compared with a parameter $pro \in (0, 1)$. If $rnd_j \leq pro$, then $New_Chromosome^j$ is set to a randomly generated real number uniformly distributed in the legal range $[L_j, U_j]$, where L_j and

U_j are the lower and upper bounds respectively of the dimension j . Otherwise, $New_Chromosome^j$ is inherited from any member selected randomly from the best $\frac{p}{2}\%$ of the population, that is to say,

$$New_Chromosome^j = \begin{cases} rand(L_j, U_j), & rnd_j \leq pro, \\ Sorted_Population_{ran}^j, & otherwise. \end{cases}$$

where $Sorted_Population$ is the sorted array containing the present population and $New_Chromosome$ has each member of the newly generated sub-population (which is of the same size as the $\frac{p}{2}\%$ of the population to be annihilated). Here, ran is a random positive integer less than the ceiling of $\frac{p}{2}\%$ of the population. Now, after generation of the desired number of chromosomes, this sub-population is injected into the present population replacing the worst $\frac{p}{2}\%$ of its members. The parameter pro is kept equal to $\frac{D}{NP}$ where D is the dimensionality of the optimization problem at hand and NP is the number of individuals in the population. This newly generated sub-population possesses some desirable features of the present population as some of its components are chosen from the best vectors of the present population.

For unimodal functions, stagnation rarely occurs; so the stagnate parameter would be ideally quite high for such functions. However, for multimodal and hybrid functions with noise, stagnation occurs pretty frequently and choosing $stagnate$ to be at a lesser value will lead to better results. Experiments with a wide range of problems suggest that $stagnate = \frac{D}{2}$ seems to be a suitable choice for black-box optimization problems.

4 Experiments and Results

4.1 Numerical Benchmarks

The proposed algorithm (abbreviated as CAR_DE from now on) is tested using a set of standard benchmark functions from the special session and competition on real parameter optimization held under the IEEE CEC 2005. These functions include a diverse set of features like multimodality, ruggedness, noise in fitness, ill-conditioning, rotation etc. and based on the classical benchmarks like Rosenbrock's, Rastrigin's, Griewank's, Schwefel's and Ackley's functions. A detailed description of these functions appears in [11] and is not repeated here. In summary, functions 1 to 5 are unimodal, functions 6 to 14 are multimodal and functions 15 to 25 are hybrid functions.

4.2 Algorithms Compared and Parametric Setup

The performance of CAR_DE is compared with the following algorithms that include two classical DE variants and four state-of-the-art adaptive DE variants:

1. DE/current-to-best/1/bin with $F = 0.8$ and $Cr = 0.9$;
2. DE/rand/1/bin with $F = 0.8$ and $Cr = 0.9$;

Table 1. Mean and Standard Deviation of the Error Values for F_1 to $F_{10}(50D)$. The Best Entries are Marked in Boldface.

| Functions → | f_1 | f_2 | f_3 | f_4 | f_5 |
|------------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Algo ↓ | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) |
| DE/rand/1/bin | 9.6015e - 05 | 3.960e + 13 | 5.469e + 7 | 1.180e + 4 | 8.709e + 03 |
| | (1.2837e - 05) | (9.307e + 02) | (1.328e + 07) | (3.332e + 03) | (6.938e + 02) |
| DE/current-to-best/bin | 2.1443e - 06 | 2.136e + 03 | 1.030e + 07 | 8.5675e + 03 | 7.462e + 03 |
| | (6.1254e - 06) | (1.128e + 03) | (6.375e + 06) | (2.8624e + 03) | (1.326e + 03) |
| JADE | 7.4615e - 14 | 5.6310e - 04 | 8.7156e+04 | 3.160e + 03 | 3.055e + 03 |
| | (2.4190e - 04) | (7.8233e - 06) | (3.6847e+04) | (4.134e - 01) | (5.485e + 02) |
| jDE | 3.1544e - 09 | 5.202e + 03 | 2.977e + 07 | 1.0194e + 04 | 4.206e + 03 |
| | (4.9946e - 09) | (1.486e + 03) | (5.744e + 06) | (2.1828e - 01) | (5.088e + 02) |
| SaDE | 1.4872e - 11 | 2.280e - 03 | 7.179e + 05 | 9.778e + 04 | 5.992e + 03 |
| | (2.8335e - 12) | (8.545e - 03) | (1.007e + 06) | (9.835e + 01) | (4.464e + 02) |
| DEGL | 2.3462e - 20 | 1.1757e - 07 | 2.3114e + 05 | 1.5746e + 03 | 5.0692e + 02 |
| | (5.6234e - 20) | (6.5592e - 08) | (1.032e + 05) | (9.501e + 00) | (5.803e + 02) |
| CAR_DE | 5.8927e-36 | 5.1189e-13 | 8.5215e + 04 | 1.5795e-02 | 4.0927e+02 |
| | (0.0000e+00) | (1.4956e-14) | (1.6874e + 04) | (2.2227e-02) | (1.6839e+02) |

| Functions → | f_6 | f_7 | f_8 | f_9 | f_{10} |
|------------------------|---------------------|---------------------|---------------------|--------------------|--------------------|
| Algo ↓ | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) |
| DE/rand/1/bin | 4.9162e + 01 | 6.1953e + 03 | 2.1142e + 01 | 3.468e + 02 | 3.763e + 02 |
| | (1.182e + 01) | (4.594e - 12) | (3.330e - 02) | (1.199e + 02) | (1.578e + 01) |
| DE/current-to-best/bin | 1.0782e + 07 | 6.6691e + 03 | 2.1133e + 01 | 2.406e + 02 | 2.5467e + 02 |
| | (1.237e + 07) | (1.795e + 02) | (3.251e - 02) | (2.939e + 01) | (7.6634e + 01) |
| JADE | 1.5413e + 01 | 6.1932e + 03 | 2.1136e + 01 | 1.352e+02 | 1.935e + 02 |
| | (1.0642e + 01) | (1.840e + 00) | (3.251e - 02) | (2.591e+00) | (2.060e + 01) |
| jDE | 4.1758e + 01 | 6.3114e + 03 | 2.1132e + 01 | 1.716e + 02 | 1.9597e + 02 |
| | (8.910e + 00) | (1.596e + 01) | (3.807e - 02) | (1.409e + 01) | (5.6236e + 01) |
| SaDE | 1.1337e + 01 | 6.1951e + 03 | 2.1132e + 01 | 1.148e + 02 | 6.342e+01 |
| | (1.044e + 01) | (4.594e - 12) | (3.458e - 02) | (1.266e + 01) | (1.287e+01) |
| DEGL | 1.3452e + 01 | 6.1953e + 03 | 2.1131e + 01 | 1.620e + 02 | 1.0217e + 02 |
| | (1.108e + 01) | (4.594e - 12) | (3.917e - 02) | (1.743e + 01) | (3.5590e + 01) |
| CAR_DE | 1.0653e+01 | 1.1084e-12 | 2.1131e+01 | 1.5422e + 02 | 1.7213e + 02 |
| | (3.9319e+00) | (4.0164e-14) | (2.0287e-02) | (2.1849e + 01) | (4.3521e + 01) |

- JADE with $c = 0.1$, $p = 0.05$, and optional external archive [12];
- jDE with $F_l = 0.1$, $F_u = 0.9$, and $\tau_1 = \tau_2 = 0.1$ [1];
- SaDE [7];
- DEGL/SAW [3] with $\alpha = \beta = F = 0.8$, $Cr = 0.9$, and neighborhood size=0.1 * NP.

The population size NP of all the DE variants is kept at 100 irrespective of the dimension D .

4.3 Simulation Strategies

Functions f_1 to f_{25} are tested for 50 and 100 dimensions. The maximum number of FEs are set to 500000 for 50D problems and 1000000 for 100D problems as per the guidelines of the CEC 2005 special session [11]. All the simulations have been done on an Intel Core-i5 CPU machine with 4 GB memory and 2.5 GHz speed.

4.4 Results on the Numerical Benchmarks

Tables 1, 2, 3 and 4 show the mean and standard deviation of the 50 best-of-the-run-errors for 50 independent runs of each of the seven algorithms on

Table 2. Mean and Standard Deviation of the Error Values for F_{11} to $F_{25}(50D)$. The Best Entries are Marked in Boldface.

| Functions → | f_{11} | f_{12} | f_{13} | f_{14} | f_{15} |
|------------------------|---------------------|--------------------|---------------------|---------------------|---------------------|
| Algo ↓ | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) |
| DE/rand/1/bin | 7.264e + 01 | 2.049e + 06 | 3.596e + 01 | 2.339e + 01 | 5.090e + 02 |
| | (1.212e + 00) | (5.925e + 05) | (1.446e + 00) | (1.486e - 01) | (7.981e + 01) |
| DE/current-to-best/bin | 4.950e + 01 | 2.505e + 05 | 3.412e + 01 | 2.286e + 01 | 4.989e + 02 |
| | (4.4551e + 00) | (1.137e + 05) | (8.9042e + 00) | (4.091e - 01) | (5.001e + 01) |
| JADE | 6.208e + 01 | 1.768e + 05 | 2.3112e + 01 | 2.284e + 01 | 3.769e + 02 |
| | (1.744e + 00) | (7.105e + 04) | (4.784e - 01) | (2.5486e - 01) | (8.764e + 01) |
| jDE | 7.330e + 01 | 1.473e + 05 | 2.5603e + 01 | 2.309e + 01 | 4.000e + 02 |
| | (1.008e + 00) | (1.928e + 05) | (1.322e + 00) | (2.8437e - 01) | (0.000e + 00) |
| SaDE | 6.634e + 01 | 8.871e+03 | 2.771e + 01 | 2.284e + 01 | 3.8827e + 01 |
| | (1.485e + 00) | (7.092e+03) | (4.112e + 00) | (2.0634e - 01) | (1.0755e + 02) |
| DEGL | 6.290e + 01 | 5.781e + 04 | 3.063e + 01 | 2.262e + 01 | 3.8982e + 02 |
| | (1.1360e + 01) | (4.566e + 04) | (4.361e + 00) | (3.3750e - 01) | (4.9284e + 01) |
| CAR_DE | 4.1575e+01 | 1.4989e + 06 | 1.4648e+01 | 2.1956e+01 | 3.6446e+02 |
| | (1.4847e+00) | (2.8857e + 05) | (4.0440e+00) | (1.0006e+00) | (1.2395e+01) |

| Functions → | f_{16} | f_{17} | f_{18} | f_{19} | f_{20} |
|------------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Algo ↓ | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) |
| DE/rand/1/bin | 2.7343e + 02 | 3.7286e + 02 | 9.9043e + 02 | 9.4100e + 02 | 9.8536e + 02 |
| | (1.0498e + 01) | (3.1287e + 01) | (4.8709e + 01) | (3.8003e + 01) | (4.4956e + 01) |
| DE/current-to-best/bin | 2.5387e + 02 | 2.5234e + 02 | 9.2089e + 02 | 9.2667e + 02 | 9.3586e + 02 |
| | (1.4757e + 01) | (5.7296e + 01) | (5.6632e + 01) | (5.9865e + 01) | (5.3925e + 01) |
| JADE | 1.437e + 02 | 1.896e + 02 | 9.206e + 02 | 9.6031e + 02 | 9.8672e + 02 |
| | (5.2267e + 01) | (3.8745e + 01) | (1.893e + 00) | (2.5236e + 01) | (1.8675e + 02) |
| jDE | 2.716e + 02 | 3.059e + 02 | 9.145e + 02 | 9.2090e + 02 | 9.9121e + 02 |
| | (4.7190e + 00) | (1.163e + 01) | (3.163e + 01) | (1.0406e + 01) | (1.5365e + 01) |
| SaDE | 1.5420e + 01 | 1.934e + 02 | 9.041e + 02 | 9.3493e + 02 | 9.3167e + 02 |
| | (6.1686e + 01) | (2.9679e + 00) | (5.208e + 01) | (1.9639e + 01) | (2.0137e + 01) |
| DEGL | 1.3153e + 02 | 1.7659e + 02 | 9.6067e + 02 | 9.1430e + 02 | 9.2196e + 02 |
| | (1.9986e + 01) | (2.3653e + 01) | (2.8458e + 01) | (2.0105e + 01) | (4.5874e + 01) |
| CAR_DE | 1.2552e+02 | 1.2393e+02 | 8.4017e+02 | 8.4094e+02 | 8.3885e+02 |
| | (1.2970e+00) | (1.3762e+00) | (1.0514e+00) | (3.1157e+00) | (1.8098e+00) |

| Functions → | f_{21} | f_{22} | f_{23} | f_{24} | f_{25} |
|------------------------|---------------------|---------------------|---------------------|------------------|---------------------|
| Algo ↓ | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) |
| DE/rand/1/bin | 9.1108e + 02 | 9.9463e + 02 | 9.1185e + 02 | 7.9849e + 02 | 1.7586e + 03 |
| | (5.7474e + 02) | (1.3465e + 01) | (2.5986e + 01) | (2.4586e + 01) | (4.5365e + 00) |
| DE/current-to-best/bin | 8.9465e + 02 | 9.3443e + 02 | 9.2645e + 02 | 7.9456e + 02 | 1.7846e + 03 |
| | (1.7465e + 02) | (1.1026e + 01) | (2.5986e + 02) | (1.3642e + 01) | (6.7654e + 00) |
| JADE | 8.523e + 02 | 9.1370e + 02 | 8.103e + 02 | 2.000e + 02 | 1.6632e + 03 |
| | (3.5175e + 02) | (2.4356e + 01) | (2.4572e + 02) | (0) | (5.5842e + 00) |
| jDE | 8.0619e + 02 | 9.796e + 02 | 8.3044e + 02 | 2.000e + 02 | 1.728e + 03 |
| | (1.0896e + 02) | (1.4851e + 01) | (1.0787e + 02) | (0) | (6.2562e + 00) |
| SaDE | 8.6400e + 02 | 9.7245e + 02 | 8.6405e + 02 | 2.000e + 02 | 1.7586e + 03 |
| | (1.5799e + 02) | (3.3383e + 01) | (1.5266e + 02) | (0) | (3.1453e + 00) |
| DEGL | 8.3600e + 02 | 9.4242e + 02 | 8.3934e + 02 | 7.2465e + 02 | 1.571e + 03 |
| | (2.1772e + 02) | (3.5647e + 01) | (1.6620e + 02) | (8.3066e + 01) | (6.5096e + 00) |
| CAR_DE | 7.3459e+02 | 5.001e+02 | 7.0374e+02 | 2.000e+02 | 2.3126e+02 |
| | (8.9992e+00) | (8.2260e+00) | (7.8156e+01) | (0) | (6.4983e+00) |

25 numerical benchmarks for 50D and on the first 14 benchmarks for 100D respectively. Note that the best-of-the-run error corresponds to the absolute difference between the best-of-the-run value $f(\mathbf{X}_{best})$ and the actual optimum f^* of a particular objective function, i.e., $|f(\mathbf{X}_{best}) - f^*|$. Table 1 and 2 reveal that CAR_DE outperformed the other DE variants in 21 out of 25 functions for 50D problems and performs well specially for multimodal and hybrid functions. Tables 3 and 4 show that this algorithm defeats the other DE variants in 11 out of 14 functions for 100D problems. Thus the increasing dimensionality does not worsen the performance of this algorithm. Note that CAR_DE retains its

Table 3. Mean and Standard Deviation of the Error Values For F_1 To $F_{10}(100D)$. The Best Entries are Marked in Boldface.

| Functions → | f_1 | f_2 | f_3 | f_4 | f_5 |
|------------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Algo ↓ | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) |
| DE/rand/1/bin | 2.9467e - 05 | 8.9371e + 04 | 9.7635e + 07 | 3.0937e + 05 | 1.0675e + 06 |
| | (3.0947e - 04) | (7.4925e + 04) | (8.4625e + 03) | (4.9875e + 03) | (4.8437e + 03) |
| DE/current-to-best/bin | 8.4735e - 06 | 9.0927e + 03 | 9.8525e + 06 | 7.5821e + 04 | 5.0967e + 05 |
| | (9.4927e - 05) | (8.4736e - 05) | (5.0948e + 03) | (8.6745e + 03) | (5.8453e + 03) |
| JADE | 6.4825e - 10 | 3.4923e + 03 | 2.9371e+06 | 5.0342e+04 | 7.5251e + 05 |
| | (4.0249e - 09) | (8.4725e - 06) | (7.4728e+04) | (6.9785e-03) | (3.9464e + 03) |
| jDE | 7.4627e - 07 | 5.9371e + 03 | 8.9372e + 06 | 7.9261e + 04 | 2.9361e + 05 |
| | (3.9371e - 09) | (8.4625e - 07) | (3.0927e + 04) | (2.7456e + 03) | (6.4536e + 03) |
| SaDE | 8.2615e - 08 | 2.9471e + 04 | 7.9171e + 06 | 6.0283e + 04 | 9.7364e + 05 |
| | (5.0445e - 09) | (7.7352e - 03) | (8.0936e + 02) | (6.9573e + 03) | (2.0936e + 03) |
| DEGL | 9.6844e - 07 | 8.9261e + 04 | 4.8326e + 07 | 1.9372e + 05 | 9.0936e + 05 |
| | (4.0937e - 08) | (8.4563e - 06) | (5.0945e + 04) | (4.9463e + 04) | (6.9382e + 03) |
| CAR_DE | 9.099e-13 | 1.5207e+01 | 4.5832e + 06 | 9.7403e + 04 | 1.7284e+04 |
| | (7.1901e-14) | (2.8769e+01) | (2.9612e + 04) | (2.6157e + 04) | (2.7346e+03) |

| Functions → | f_6 | f_7 | f_8 | f_9 | f_{10} |
|------------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Algo ↓ | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) |
| DE/rand/1/bin | 1.8946e + 05 | 1.8574e + 05 | 2.2497e + 01 | 9.3752e + 02 | 9.7631e + 02 |
| | (4.9375e + 01) | (4.0832e + 02) | (3.0423e + 01) | (6.7322e + 01) | (1.5787e + 01) |
| DE/current-to-best/bin | 9.3967e + 04 | 1.6738e + 05 | 2.2308e + 01 | 9.4065e + 02 | 7.5467e + 02 |
| | (7.4925e + 01) | (8.4735e + 02) | (8.4725e + 00) | (7.6695e + 01) | (5.64334e+01) |
| JADE | 7.5329e + 04 | 9.0417e + 04 | 2.1964e + 01 | 8.9272e + 02 | 5.935e + 02 |
| | (7.3725e + 01) | (8.4735e + 02) | (9.4673e + 01) | (4.5043e + 01) | (8.8760e + 01) |
| jDE | 8.1876e + 04 | 1.1625e + 05 | 2.2197e + 01 | 9.3264e + 02 | 6.9597e + 02 |
| | (8.4752e + 00) | (7.4637e + 02) | (7.4627e + 00) | (7.3645e + 01) | (7.7653e + 01) |
| SaDE | 2.0172e + 04 | 9.8463e + 04 | 2.2075e + 01 | 9.2737e + 02 | 8.3426e + 02 |
| | (8.2514e + 01) | (8.4623e + 02) | (9.3736e + 00) | (5.1369e + 01) | (1.6378e + 01) |
| DEGL | 4.2684e + 04 | 1.2383e + 05 | 2.2210e + 01 | 9.5591e + 03 | 8.2017e + 02 |
| | (5.8352e + 01) | (9.4573e + 02) | (4.8252e + 00) | (3.9274e + 01) | (3.5590e + 01) |
| CAR_DE | 1.8396e+02 | 1.8914e+04 | 2.1674e+01 | 5.9232e+02 | 4.9012e+02 |
| | (9.8705e+01) | (1.5657e+03) | (3.0845e+01) | (2.2333e+01) | (8.4560e+01) |

Table 4. Mean and Standard Deviation of the Error Values for F_{11} to $F_{14}(100D)$. The Best Entries are Marked in Boldface.

| Functions → | f_{11} | f_{12} | f_{13} | f_{14} |
|------------------------|---------------------|---------------------|---------------------|---------------------|
| Algo ↓ | Mean(Std) | Mean(Std) | Mean(Std) | Mean(Std) |
| DE/rand/1/bin | 9.2648e + 01 | 9.049e + 06 | 5.2926e + 01 | 4.4339e + 01 |
| | (1.9212e + 00) | (7.925e + 05) | (4.346e + 00) | (1.4836e - 01) |
| DE/current-to-best/bin | 6.9150e + 01 | 6.505e + 05 | 4.4132e + 01 | 4.5186e + 01 |
| | (6.6451e + 00) | (4.137e + 05) | (3.5042e + 00) | (4.4091e - 01) |
| JADE | 9.1208e + 01 | 6.768e + 05 | 3.4141e + 01 | 4.2684e + 01 |
| | (4.7744e + 00) | (2.105e + 04) | (4.0834e + 00) | (3.8163e - 01) |
| jDE | 7.7630e + 01 | 7.473e + 05 | 3.8650e + 01 | 4.3309e + 01 |
| | (9.4008e + 00) | (7.928e + 05) | (4.322e + 01) | (2.7163e - 01) |
| SaDE | 8.4684e + 01 | 9.0781e+04 | 3.5713e + 00 | 4.2874e + 01 |
| | (3.485e + 00) | (5.9092e+03) | (5.112e + 00) | (2.5343e - 01) |
| DEGL | 8.1290e + 01 | 8.8781e + 05 | 4.063e + 01 | 4.2662e + 01 |
| | (1.360e + 01) | (7.1566e + 04) | (2.361e + 01) | (6.9834e - 01) |
| CAR_DE | 6.2910e+01 | 4.4677e + 05 | 3.3179e+01 | 4.0225e+01 |
| | (4.1313e+00) | (1.8670e + 04) | (4.9355e+00) | (2.4122e-01) |

superiority in case of rotated, shifted functions and functions with noise in fitness. Therefore, function rotation and incorporation of multiplicative noise does not hamper the performance of the algorithm significantly.

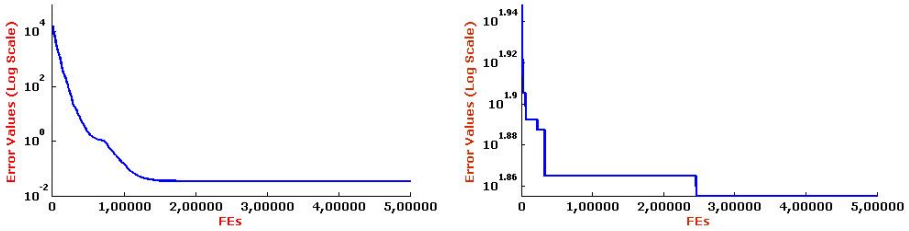


Fig. 1. Progress toward the optimum solution for median run of eight algorithms over two numerical benchmarks (in 50D). Left: Shifted rotated Griewank's function f_7 ; Right: Shifted Rotated Weierstrass function f_{11} .

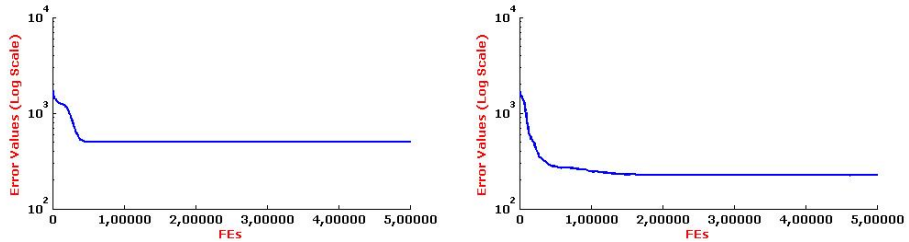


Fig. 2. Progress toward the optimum solution for median run of eight algorithms over two numerical benchmarks (in 50D). Left: Rotated Hybrid Composition Function f_{21} ; Right: Composition function f_{25} .

For further illustration, in Fig. 1 and 2, we show the convergence graphs for the median run of (where the runs were sorted according to the final best error values achieved in each) the CAR_DE algorithm on four benchmarks in 50D.

5 Conclusion

The results for 50 dimensions reveal the performance of CAR_DE to be almost comparable to that of JADE (and SADE in some instances) for all the unimodal functions, however the performance of CAR_DE is slightly better for multimodal functions, and significantly better for hybrid functions. In all the hybrid functions, CAR_DE performs better than all the state-of-the-art DE variants. The results obtained for higher dimensions support the superiority of CAR_DE over the other DE variants. Specially, the hybrid functions of 50 dimensions and multimodal functions of 100 dimensions lucidly demonstrate how the novel mutation scheme coupled with annihilation and regeneration can improve the efficiency of DE.

The real challenge to an evolutionary algorithm comes in the form of optimization of highly complex, noisy and higher dimensional functions (such as those working in 50 or 100 dimensions). As the complexity of the function increases, there arrives the need to efficiently explore the search space.

The present algorithm tries to optimize such more complex functions, attaining that purpose without losing much of its explorative nature for functions

of smaller dimensions. Further improvements to the CAR_DE algorithm may be made by tuning the algorithmic parameters and maintaining an adaptive external archive.

References

1. 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 Trans. Evol. Comput.* 10(6), 646–657 (2006)
2. Das, S., Suganthan, P.N.: Differential Evolution A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* 15(1), 4–31 (2011)
3. Das, S., Abraham, A., Chakraborty, U.K., Konar, A.: Differential evolution using a neighbourhood based mutation operator. *IEEE Trans. Evol. Comput.* 13(3), 526–553 (2009)
4. Islam, S.M., Das, S., Ghosh, S., Roy, S., Suganthan, P.N.: An Adaptive Differential Evolution Algorithm With Novel Mutation and Crossover Strategies for Global Numerical Optimization. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics* 42(2), 482–500 (2012)
5. 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)
6. Price, K.V., Storn, R., Lampinen, J.: *Differential Evolution A practical Approach to Global Optimization*. Springer, Berlin (2005)
7. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential Evolution Algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.* 13(2), 398–417 (2009)
8. Storn, R., Price, K.V.: Differential Evolution A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* 11(4), 341–359 (1997)
9. Storn, R., Price, K.V.: *Differential Evolution A simple and efficient adaptive scheme for global optimization over continuous spaces*, ICSI, Berkeley, CA. Tech. Rep. TR-95-012
10. Storn, R., Price, K.V.: Minimizing the real functions of the ICEC 1996 contest by differential evolution. In: *Proc. IEEE Int. Conf. Evol. Comput.*, Nagoya, Japan, pp. 842–844 (1996)
11. 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 Technol. Univ., Singapore (2005)
12. Zhang, J., Sanderson, A.C.: JADE: Adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* 13(5), 945–958 (2009)

Differential Evolution and Offspring Repair Method Based Dynamic Constrained Optimization

Kunal Pal¹, Chiranjib Saha¹, and Swagatam Das²

¹ Electronics and Telecommunication Engineering, Jadavpur University, Kolkata-32

² Electronics and Communication Science Unit, Indian Statistical Institute, Kolkata

Abstract. Most of the real world optimisation problems are inherently dynamic and constrained. In a Dynamic Constrained Optimization Problem (DCOP), the objective function as well as the constraint functions change with respect to time. While several algorithms already exist in the purview of dynamic optimization, the introduction of constraint makes the challenge more sophisticated. Conventional DCO algorithms involve a Core-Optimizer (e.g. GA, PSO etc.) accompanied by a separate constraint-handling technique e.g., a repair method, or a penalty function. However, it has been observed that ordinary repair methods with elitism significantly decrease the diversity of the population during the exploitation stage and the penalty functions cannot properly deal with disconnected feasible regions. In this paper, we present a new algorithm based on the Differential Evolution algorithm as well as a modified version of a repair method that produces improved results. The proposed approach incorporates knowledge-reusing and knowledge-restarting in order to produce a quick recovery and faster convergence.

1 Introduction

Basic goal of constrained optimization problems [1], is to minimize the objective function subjected to few constraints. Evolutionary Algorithms (EAs) are efficient to perform unconstrained search. But in constrained environments, the search process becomes complicated due to additional criterion of satisfying the constrains and often these two objectives become conflicting.

Dynamic Optimization problems (DOPs) are a large subset of the real world optimization problems where the objective function is dynamic in nature i.e., the objective function varies with time. These problems are different from dynamic problems (also known as dynamic environments) only when *the underlying fitness landscape changes during the operation of the EA* [2]. These problems are to be solved online by an optimization algorithm as time goes on. The Dynamic Constrained Optimization Problems, as referred in [3] are more challenging, since the objective function as well as the constraints changes with time. An efficient DCO algorithm, therefore must be able to sense the change in the optimization environment and adaptively modify its search strategy.

The prime challenges of DCOP can be shortlisted in two categories i.e. outdated memory and lack of diversity. Usually after a change in the environment, the intelligence gathered by the population becomes outdated. So the algorithm must be able to track change and to do that the population should contain enough diversity to explore the problem landscape.

Recent works on DCOP has involved the triggered Hyper-Mutation Genetic Algorithm. In hyper-mutation, introduced in [4], the mutation rates are triggered to a very large value whenever a change is detected. In this way, this algorithm copes with the dynamics of the environment by introducing diversity in the population and then the algorithm adapts to new environments. However, this algorithm cannot detect a change when new global optima are exposed without changing the objective value of the previous optima. Also, determining the correct mutation rate is very difficult since it is a function of the degree of change in the environment.

In the paradigm of swarm based optimization algorithms, Particle Swarm Optimization, applied by Hu and Eberhart [7], was modified such that the swarm is diversified by randomized relocation of the particles after a change in the environment is detected. Following a more sophisticated approach [8], the swarm can be divided into a hierarchy of sub-swarms in order to introduce diversity. But, determining the number of sub-swarms is a tough challenge and this algorithm fails to converge to an optimum when there is a large change in the objective landscape, as well.

This paper introduces an algorithm to solve dynamic constrained optimization problems efficiently using the DE with an offspring repair technique. The proposed algorithm tracks changes in the environment separately and relocates the population all over the modified feasible region whenever environment is re-structured. Thus, it can search the whole feasible space and explore new global optima irrespective of the presence of small or large changes in the fitness space. The worst particle in every generation is randomly replaced by a new particle from the feasible space which introduces diversity in the algorithm. Also, the gravitational constant is replaced by a high value whenever the population is relocated and henceforth, maintains diversity for the first few generations.

Of the CH methods available, the penalty function, which is the most popular CH method in current use, fails when there are several disconnected feasible regions in the constraint space. Further dynamic feasible regions and switching of the global optimum amongst the regions separated by high degree of infeasibility aggravates the situation. The CH technique used here is the repair method. In this technique, if any offspring enters the infeasible region, it is replaced by a random particle between it and its closest feasible solution in the population. Therefore, the information in the infeasible particle is partially preserved.

A short introduction to the Dynamic Constrained Optimization Problem is provided in Section 2. This section also discusses some real-life applications of DCOPs. Section 3 introduces Differential Evolution (DE) [9]. The shortcomings of conventional CH techniques and the benefits of using a repair method are discussed in Section 4. Section 5 describes the detection of change in an

environment and Section 6 shows a comparative analysis of several DCO algorithms. This section also gives a thorough discussion explaining the better performance of our algorithm. Finally, Section 7 provides our conclusions and suggests some possible paths for future research.

2 Dynamic Constrained Optimization

Without loss of generality, a Dynamic Constrained Optimization Problem in continuous domain can be stated as follows:

$$\min_{x \in D_t \subseteq [L, U]} f(x, t) \quad (1)$$

$$G_i(x, t) \leq 0, \forall i \in \{1, 2, \dots, m\}, \quad (2)$$

where $t \in N^+$ is called time (environment) variance, $[L, U] = \{x = (x_1, x_2, \dots, x_n) \mid L_i \leq x_i \leq U_i, i = 1 \sim n\}$ is called search space and $x = \{x \mid x \in [L, U], G_i(x, t) \leq 0, i = 1 \sim m\}$ is called feasible space.

For $\forall x \in D_t$, if there exists a point $x^* \in D_t$ such that $f(x^*, t) \leq f(x, t)$ (or $f(x^*, t) \geq f(x, t)$ for maxima) then x^* is called the optimal point and $f(x^*, t)$ is called the optimal value for the environment t .

There are miscellaneous real life applications for continuous DCOP from which optimal control of Hybrid systems [10,11] and Dynamic systems [12] constitute more than 70%. Other than that, source identification, parameter estimation, pattern recognition and classification are other popular known fields of application for DCOP [1].

3 Differential Evolution

Differential evolution (DE) has emerged as one of the most competitive evolutionary algorithm. Invented by Storn and Prince, DE is a stochastic direct search method using population or multiple search points [34]. Its variants have been successfully implemented in solving multi-objective, dynamic and constrained optimization problems and tackle many real world situations.

DE has the structure similar to genetic algorithm. It is equipped with a population which is a collection of trial solutions. In case of real parameter optimisation, the parameters to be optimised are encoded with in a vector $X = [x_1, x_2, \dots, x_n]$. These individual vectors (which constitute a population) are called parameter vectors or genomes. The population is then manipulated with three operations namely mutation, crossover and selection. However, unlike traditional EAs, DE employs difference of the parameter vectors to explore the objective function landscape. It perturbs the population members with the scaled differences of randomly selected and distinct population members. Therefore, there exists no separate probability distribution for generating the offspring.

In the search population, each vector forms a candidate solution to the multi-dimensional optimization problem. We shall denote subsequent generations in DE by $G = 0, 1, \dots, G_{max}$. Since the parameter vectors are likely to

be changed over different generations, we may adopt the following notation for representing the i^{th} vector of the population at the current generation $X_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \dots, x_{D,i,G}]$. Where $x_{j,i,0}$ is randomly initialised with in the search space constrained by the prescribed minimum and maximum bounds: $x_{j,min}$ and $x_{j,max} \forall j = 1, 2, \dots, N$. Hence we may initialize the j^{th} component of the i^{th} vector as

$$x_{j,i,0} = x_{j,min} + rand \times (x_{j,max} - x_{j,min}), \tag{3}$$

where $rand$ is a uniformly distributed random number lying between 0 and 1 and is instantiated independently for each component of the i^{th} vector.

Mutation, in the paradigm of DE, signifies a random perturbation about a trail vector [34]. In the simplest version of mutation, three non overlapping vectors $X_{p_1,i,G}^i, X_{p_2,i,G}^i, X_{p_3,i,G}^i$ are randomly selected from the population (p_1^i, p_2^i, p_3^i are three mutually exclusive integers belonging to the range $[1, NP]$). The donor vector which is the outcome of mutation is generated as,

$$V_{i,G} = X_{p_1,i,G}^i + F \times (X_{p_2,i,G}^i - X_{p_3,i,G}^i) \tag{4}$$

where F is the scaling factor and $F \in [0.41]$.

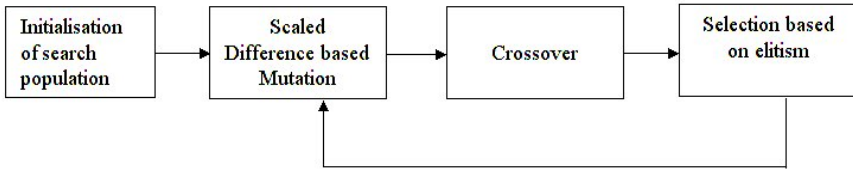


Fig. 1. Differential Evolution

After generating the donor vector, the cross over operation comes into play to enhance the potential diversity of the population. The genes of $V_{i,G}$ and $X_{i,G}$ are interchanged to form a trial vector $U_{i,G} = [u_{1,i,G}, u_{2,i,G}, \dots, u_{D,i,G}]$. The scheme of binomial crossover is dictated by the following.

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if } rand \leq Cr \text{ or } j = j_{rand} \\ x_{j,i,G}, & \text{otherwise} \end{cases} \tag{5}$$

where $j_{rand} \in [1, 2, \dots, NP]$ is a randomly chosen index ensuring that at least one component of $U_{i,G}$ is selected from $V_{i,G}$.

4 Handling Constraints

Conventional CH techniques [35] such as penalty functions are inefficient for solving DCOPs since they are unable to track the optima when the feasible region is disconnected and the optimum switches between these regions. On the

contrary, a repair method maps infeasible individuals into feasible solutions with the help of a reference population.

Repair method was introduced by Michalewicz and Nazhiyath [13]. The main advantage of a repair method over conventional CH techniques is its independence of the amount of constraint violation and its ability to preserve even an infeasible individual which has the capability of producing good solution. On the other hand, it has been observed that repair methods are, in a way, very robust, and therefore, they fulfil the most significant requirements for dynamic constraint handling.

In a repair method, we generate first a reference population which entirely belongs to the feasible region. Now, if S is an infeasible individual, then a random point is generated on the straight line joining S and any member R from the reference population. If the point belongs to the feasible region, then it replaces S in the main population. The reference population is updated as well if the new solution has a better fitness value than the member from the reference population.

This conventional repair method has been modified in our proposed algorithm. Here, the initial population is generated entirely in the feasible region—i.e., the size of the reference population is the same as the size of the overall population. Therefore, the main population here is a null set.

Now, if the standard deviation of the population is low, then the repair method significantly decreases the diversity of the population. Therefore, the exploration process is affected. To get rid of this phenomenon, we have used a modified version of the repair method. Instead of selecting R randomly selected individuals from the reference population, we choose R individuals such that the distance between R and S is minimum. In this way, the information contained in the infeasible particle is not lost and a feasible solution is also produced. The random selection of the generated particle also gives the mechanism a heuristic touch.

5 Detection of Changes in Environment

DCO differs from its static counterpart by the fact that either the objective function or the constraints (or both) keep changing with time. In the benchmark problems in this area [3], the time dependence has been modelled by setting a number of function evaluations (1000 objective function evaluations) upto which the environment is static. Henceforth, our problem can be viewed as a series of distinct static constrained optimisation problems. However, in practical situations, the environment can change with any random frequency. Considering the worst case scenario to be when the environment changes very rapidly, the optimization algorithm must be able to work very fast. To achieve that, the best alternative is to use the knowledge from the previous search and that is why this algorithm uses knowledge-reuse as well as knowledge-regeneration. Now, if there is a change in the environment and it remains undetected, then the algorithm is said to fail to cope with the dynamic nature of the problem. So, the detection of change in the environment is very important and must be very fast.

The detection of changes is realized with one or a fixed set of detectors and their present and past objective values and constraint violations are compared. If they are different, then, indeed, the environment has changed. Average best fitness or some members from the population or a point (which could be random or fixed) can be used as detectors. However, if the algorithm is assisted with elitism, then the diversity of the population decreases eventually with generations. So, the population is trapped in a fixed part of the fitness landscape and if the change is not in that part, then there is a possibility that the change will be left undetected. So, this algorithm uses an entirely isolated set of detectors and compromises some objective function evaluations to achieve a quick and sensitive change detection system.

There are two types of changes that can affect the search process. One, is a change in the objective function, and the other is a change in the constraint functions. Both types of changes can be efficiently detected by establishing a set of fixed points (i.e. pivots) and then evaluating the fitness value and the amount of constraint violation after each iteration. Changes in the constraint functions not only change the boundary of the feasible region but also add (or remove) disjoint feasible regions or can change the amount of violation of any individual, while keeping the boundary of the feasible region fixed. To detect changes in the constraint landscape, we do not require any extra evaluations and, therefore, there is no limitation in this case. Hence, an arbitrary number of pivots can be assigned for detection. Now, for the former case, each time a change is detected, some evaluations are required to set the reference values for the pivots. Added to that, there will be some more evaluations required to compare the past and present values of the detectors. But, it has been observed that the number of fitness function evaluations needed for change detection is substantially small when compared to that used for optimization purposes.

6 Results and Discussions

Real-life applications of DCOPs are numerous, e.g. dynamic aircraft scheduling [14], dynamic vehicle routing [15], video based motion capture [16], modelling of oscillatory behavior of bacterial cultures [17], modelling and control of open plate reactors [18], etc. In real-world DCO problems, the objective function and constraint functions can be combined in three different types. The first type of combination is the case where both the objective function and the constraints are dynamic, as in scheduling/resource allocation problems [27], aerodynamic/structural design problems [31], or in many optimal control problems [28][29][30]. In the second type of combination, the objective function is dynamic while the constraints are static, for example, in the document stream modelling problem [32], the evolvable hardware design problem [23] or the optimal control problem of fermentation processes [24]. In the third type of combination, the objective function is static and the constraints are dynamic, as can be seen in the hydrothermal scheduling problem [25], the cargo movement problem [26] and the ship scheduling problem [33].

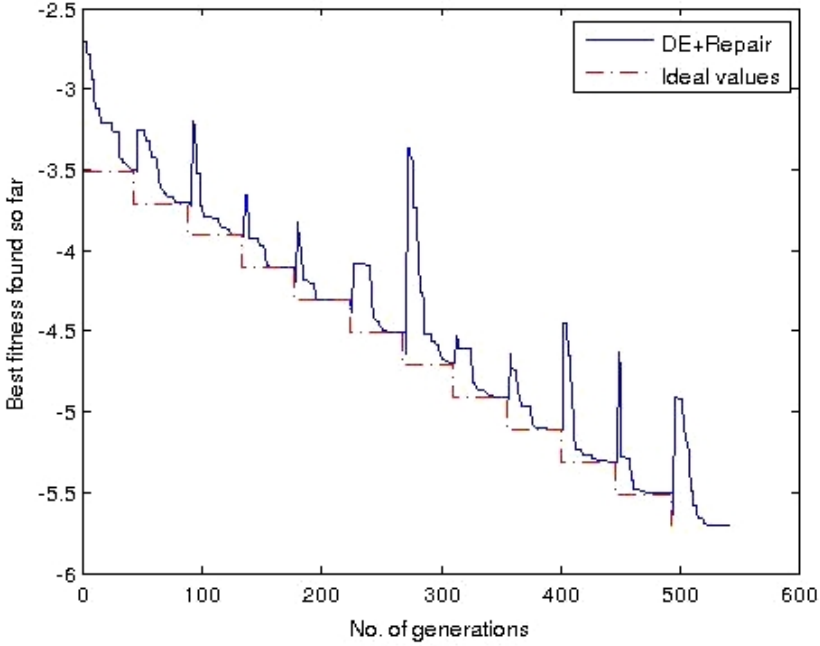


Fig. 2. $G24_3$ Fixed Objective Dynamic Constrained problem ($N = 25$)

The test environment used here contains all these three types of real world DCOPs and makes the optimization process very challenging by narrowing the feasible space sometimes e.g., in $G24_5$ and $G24_7$, the percentage of feasible region reduces linearly from 44.61% to 7.29%. Another challenge for the algorithms is the presence of disconnected feasible region e.g., $G24_3$ and $G24_3b$, represents the first and the third type of DCOPs respectively and contains 2-3 disconnected feasible regions in average. The performance analysis of the DCO algorithms are subject to the test problems adopted in this paper.

The main goal of optimization is producing the best possible solution which is reflected in this case by the mean error. Although the problems here are dynamic in nature, a better way to calculate the error as indicated in [1] is the offline error i.e., the mean of the error in all the generations. A ranking based on the offline errors are given in Table 1.

In real life problems, the change frequency may be very high. Considering the worst case scenario, therefore, the DCO algorithm must gain quick recovery after changes in the environment. Techniques such as the triggered hyper mutation genetic algorithm enhance the mutation rate to a very high level every time a change is detected in the environment. A few extra generations are still needed to amplify the diversity to a reasonable level. In our algorithm, each time that a change is detected, a part of the old population dies out and the same number of particles are regenerated. Combined with the faster convergence of DE, this technique provides a faster recovery.

Table 1. Offline errors of the tested algorithms in the medium settings (pop size=25; change frequency=1000 evaluations; objective functions change severity $k = 0.5$; constraint functions change severity $S = 20$).

| problems→ | G24-u(dF, noC) | | | G24-1(dF, fC) | | | G24-f(fF, fC) | | |
|---------------|----------------|--------------|----------|---------------|--------------|----------|---------------|--------------|----------|
| algo.↓ | mean | std | Rank | mean | (std) | Rank | mean | (std) | Rank |
| GA-noElit | 0.298 | 0.051 | 7 | 0.609 | 0.064 | 8 | 0.676 | 0.085 | 8 |
| RIGA-noElit | 0.221 | 0.025 | 6 | 0.493 | 0.045 | 7 | 0.546 | 0.072 | 7 |
| HyperM-noElit | 0.206 | 0.035 | 5 | 0.361 | 0.065 | 4 | 0.226 | 0.056 | 6 |
| GA-elit | 0.106 | 0.035 | 2 | 0.459 | 0.057 | 6 | 0.154 | 0.083 | 4 |
| RIGA-elit | 0.149 | 0.025 | 4 | 0.346 | 0.046 | 3 | 0.178 | 0.051 | 5 |
| HyperM-elit | 0.111 | 0.026 | 3 | 0.384 | 0.065 | 5 | 0.151 | 0.053 | 3 |
| GA+Repair | 0.468 | 0.059 | 8 | 0.226 | 0.024 | 2 | 0.041 | 0.011 | 2 |
| DE+Repair | 0.099 | 0.010 | 1 | 0.151 | 0.024 | 1 | 0.039 | 0.022 | 1 |

| problems→ | G24-uf(fF, noC) | | | G24-2(dF, fC) | | | G24-2u(dF,noC) | | |
|---------------|-----------------|--------------|----------|---------------|--------------|----------|----------------|--------------|----------|
| algo.↓ | mean | std | Rank | mean | (std) | Rank | mean | (std) | Rank |
| GA-noElit | 0.464 | 0.064 | 8 | 0.356 | 0.049 | 8 | 0.159 | 0.041 | 7 |
| RIGA-noElit | 0.342 | 0.032 | 7 | 0.264 | 0.035 | 5 | 0.107 | 0.019 | 4 |
| HyperM-noElit | 0.124 | 0.041 | 5 | 0.257 | 0.045 | 4 | 0.130 | 0.022 | 5 |
| GA-elit | 0.063 | 0.022 | 2.5 | 0.288 | 0.050 | 7 | 0.073 | 0.017 | 2 |
| RIGA-elit | 0.069 | 0.020 | 4 | 0.246 | 0.037 | 2 | 0.091 | 0.024 | 3 |
| HyperM-elit | 0.063 | 0.012 | 2.5 | 0.253 | 0.043 | 3 | 0.068 | 0.016 | 1 |
| GA+Repair | 0.218 | 0.018 | 6 | 0.281 | 0.036 | 6 | 0.294 | 0.029 | 8 |
| DE+Repair | 0.057 | 0.019 | 1 | 0.191 | 0.014 | 1 | 0.141 | 0.012 | 6 |

| problems→ | G24-3(fF,dC) | | | G24-3b(dF,dC) | | | G24-3f(fF, fC) | | |
|---------------|--------------|--------------|----------|---------------|--------------|----------|----------------|--------------|----------|
| algo.↓ | mean | std | Rank | mean | (std) | Rank | mean | (std) | Rank |
| GA-noElit | 0.760 | 0.099 | 8 | 0.657 | 0.097 | 8 | 0.886 | 0.179 | 8 |
| RIGA-noElit | 0.538 | 0.047 | 7 | 0.500 | 0.038 | 7 | 0.651 | 0.055 | 7 |
| HyperM-noElit | 0.411 | 0.052 | 6 | 0.459 | 0.069 | 6 | 0.256 | 0.057 | 6 |
| GA-elit | 0.289 | 0.049 | 4 | 0.457 | 0.084 | 5 | 0.158 | 0.058 | 5 |
| RIGA-elit | 0.308 | 0.048 | 5 | 0.386 | 0.051 | 3 | 0.167 | 0.048 | 3.5 |
| HyperM-elit | 0.283 | 0.050 | 3 | 0.394 | 0.088 | 4 | 0.158 | 0.051 | 3.5 |
| GA+Repair | 0.156 | 0.008 | 2 | 0.171 | 0.019 | 2 | 0.025 | 0.008 | 2 |
| DE+Repair | 0.091 | 0.012 | 1 | 0.121 | 0.019 | 1 | 0.013 | 0.009 | 1 |

| problems→ | G24-4(dF, dC) | | | G24-5(dF,dC) | | | G24-6a(dF,2DR) | | |
|---------------|---------------|--------------|----------|--------------|--------------|----------|----------------|--------------|----------|
| algo.↓ | mean | std | Rank | mean | (std) | Rank | mean | (std) | Rank |
| GA-noElit | 0.621 | 0.101 | 8 | 0.379 | 0.067 | 8 | 0.529 | 0.108 | 7 |
| RIGA-noElit | 0.490 | 0.053 | 7 | 0.293 | 0.046 | 7 | 0.366 | 0.030 | 3 |
| HyperM-noElit | 0.469 | 0.057 | 6 | 0.275 | 0.034 | 6 | 0.383 | 0.051 | 4 |
| GA-elit | 0.453 | 0.075 | 5 | 0.266 | 0.029 | 5 | 0.674 | 0.157 | 8 |
| RIGA-elit | 0.421 | 0.047 | 3 | 0.240 | 0.035 | 3 | 0.333 | 0.050 | 2 |
| HyperM-elit | 0.426 | 0.075 | 4 | 0.248 | 0.039 | 4 | 0.491 | 0.071 | 6 |
| GA+Repair | 0.211 | 0.015 | 2 | 0.236 | 0.024 | 2 | 0.431 | 0.074 | 5 |
| DE+Repair | 0.121 | 0.021 | 1 | 0.121 | 0.011 | 1 | 0.047 | 0.009 | 1 |

| problems→ | G24-6b(dF,fC,1R) | | | G24-6c(dF,2DR,easy) | | | G24-6d(dF,2DR,hard) | | |
|---------------|------------------|--------------|----------|---------------------|--------------|----------|---------------------|--------------|----------|
| algo.↓ | mean | std | Rank | mean | (std) | Rank | mean | (std) | Rank |
| GA-noElit | 0.448 | 0.054 | 8 | 0.446 | 0.041 | 8 | 0.543 | 0.127 | 8 |
| RIGA-noElit | 0.331 | 0.035 | 3 | 0.329 | 0.039 | 4 | 0.366 | 0.040 | 4 |
| HyperM-noElit | 0.340 | 0.046 | 4 | 0.323 | 0.037 | 2 | 0.370 | 0.046 | 5 |
| GA-elit | 0.408 | 0.057 | 6 | 0.441 | 0.052 | 7 | 0.510 | 0.075 | 7 |
| RIGA-elit | 0.309 | 0.039 | 2 | 0.325 | 0.029 | 3 | 0.342 | 0.057 | 2 |
| HyperM-elit | 0.390 | 0.039 | 5 | 0.394 | 0.051 | 6 | 0.456 | 0.041 | 5 |
| GA+Repair | 0.427 | 0.048 | 7 | 0.390 | 0.038 | 5 | 0.354 | 0.038 | 3 |
| DE+Repair | 0.101 | 0.012 | 1 | 0.79 | 0.010 | 1 | 0.91 | 0.011 | 1 |

| problems→ | G24-7(fF, dC) | | | G24-8a(dFnC,ONISB) | | | G24-8b(dFfC,OICB) | | |
|---------------|---------------|--------------|----------|--------------------|--------------|----------|-------------------|--------------|----------|
| algo.↓ | mean | std | Rank | mean | (std) | Rank | mean | (std) | Rank |
| GA-noElit | 0.721 | 0.088 | 8 | 0.426 | 0.050 | 8 | 0.835 | 0.068 | 8 |
| RIGA-noElit | 0.543 | 0.059 | 7 | 0.346 | 0.031 | 6 | 0.719 | 0.071 | 7 |
| HyperM-noElit | 0.495 | 0.053 | 6 | 0.374 | 0.043 | 7 | 0.681 | 0.072 | 6 |
| GA-elit | 0.316 | 0.053 | 4 | 0.266 | 0.028 | 2 | 0.662 | 0.056 | 5 |
| RIGA-elit | 0.416 | 0.068 | 5 | 0.304 | 0.028 | 5 | 0.598 | 0.064 | 3 |
| HyperM-elit | 0.315 | 0.062 | 3 | 0.279 | 0.028 | 3 | 0.608 | 0.071 | 4 |
| GA+Repair | 0.181 | 0.017 | 2 | 0.300 | 0.033 | 4 | 0.251 | 0.051 | 2 |
| DE+Repair | 0.033 | 0.009 | 1 | 0.217 | 0.033 | 1 | 0.227 | 0.039 | 1 |

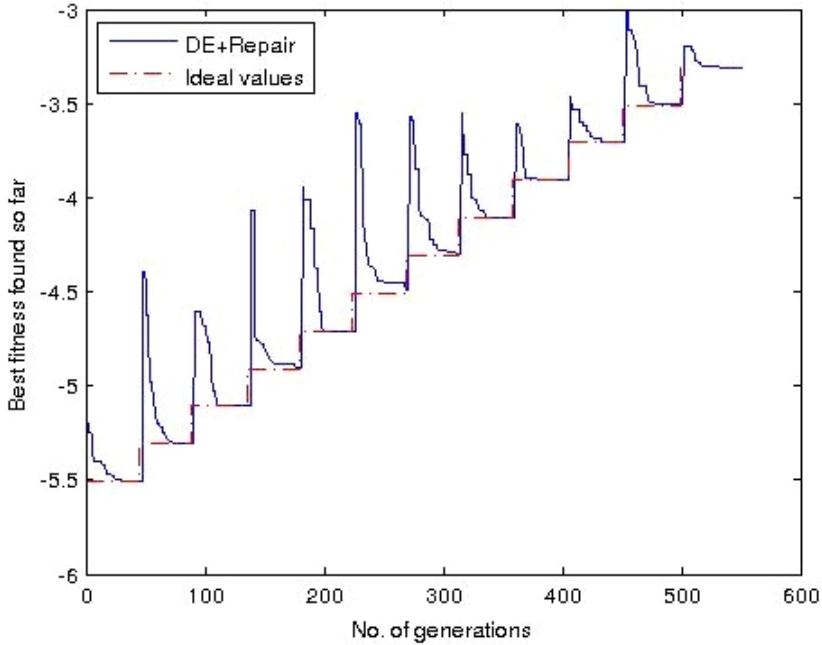


Fig. 3. $G24_7$ Fixed Objective Dynamic Constrained problem ($N = 25$)

The change detection technique used in conventional DCOPs is based on one or on a fixed set of detectors. Hence, most of the known algorithms usually track the change based on the average best fitness [19] or on some members of the population or on a point (single detector, could be random or fixed). Although, in order to get better solution, the diversity of the population should be allowed to eventually decrease and the population can be centered in a part of the objective space. If the change occurs in some other zone, then it cannot be detected using the members of the same population. To get rid of that, this algorithm is assisted with an entirely isolated detection system. The use of a single detector is risky for obvious reasons. So, we used a detector-grid made of a fixed set of equally spaced points as reference and compared for a random member from that grid. To make the method more sensitive, the number of detectors and the comparators can be increased at the expense of function evaluations.

On the other hand, some members from the previous population remain intact at every regeneration of the population. So, if the change does not affect the global optima, then the algorithm can easily deal with that, too. The best solution in this case would be not to regenerate the population, although this algorithm cannot indulge in that. But this does not change the statistics in a significant way. Thus, this algorithm combines knowledge-reusing, that produces faster solution and helps to learn the nature of the dynamics while knowledge-restarting produces better solutions.

7 Conclusions and Future Work

DCOPs are meant to be solved online. In this paper, we proposed an algorithm that combines DE and a repair technique to produce better objective function values. This algorithm divides the whole problem into some discrete cases and solves them separately while they are linked together with a change detection technique to cope up with the dynamic nature of the problem.

The modified repair method we proposed preserves diversity. It does not get trapped by disconnected feasible regions like the penalty function and performs better in handling constraints.

The change detection technique used in this algorithm is basic but very effective. Although it can be made more efficient by selecting more pivot points simultaneously, while compromising some objective function evaluations. Nevertheless this sort of scheme would have some problems when the change-frequency is very high. In the proposed algorithm, we have made a trade-off between these two issues. However, in our future work, we aim to design a different approach that consumes less function evaluations to detect a change in the environment.

References

1. Nguyen, T.: Classifying and characterising dynamic optimisation problems - a literature review. tech. rep., School of Computer Science, The University of Birmingham, UK (2007)
2. Morrison, R.W.: *Designing Evolutionary Algorithms for Dynamic Environments*. Springer, Berlin (2004) ISBN 3-540-21231-0
3. Nguyen, T.T.: A proposed real-valued dynamic constrained benchmark set. Technical report, School of Computer Science, University of Birmingham (2008a)
4. Cobb, H.G.: An Investigation into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-Dependent Nonstationary Environments. Technical Report AIC-90-001, Naval Research Laboratory, Washington, USA (1990)
5. Liu, L., Wang, D.-W., Yang, S.: Compound Particle Swarm Optimization in Dynamic Environments. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., McCormack, J., O'Neill, M., Romero, J., Rothlauf, F., Squillero, G., Uyar, A.Ş., Yang, S. (eds.) *EvoWorkshops 2008*. LNCS, vol. 4974, pp. 616–625. Springer, Heidelberg (2008)
6. Liu, C.A.: New Dynamic Constrained Optimization PSO Algorithm. In: *ICNC 2008: Proceedings of the 2008 Fourth International Conference on Natural Computation*, pp. 650–653. IEEE Computer Society (2008a)
7. Hu, X., Eberhart, R.: Adaptive particle swarm optimisation: detection and response to dynamic systems. In: *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2002*, pp. 1666–1670 (2002)
8. Janson, S., Middendorf, M.: A hierarchical particle swarm optimizer for noisy and dynamic environments. *Genetic Programming and Evolvable Machines* 7(4), 329–354 (2006)
9. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA- A Gravitational Search Algorithm. *Elsevier, Information Sciences* 179, 2232–2248 (2009)

10. de Prada, C., Sarabia, D., Cristea, S., Mazaeda, R.: Plant-wide Control of a Hybrid Process. *International Journal of Adaptive Control and Signal Processing* 22(2), 124–141 (2008)
11. Fiacchini, M., Alamo, T., Alvarado, I., Camacho, E.F.: Safety Verification and Adaptive Model Predictive Control of the Hybrid Dynamics of a Fuel Cell System. *International Journal of Adaptive Control and Signal Processing* 22(3), 142–160 (2008)
12. Dini, D., van Lent, M., Carpenter, P., Iyer, K.: Building robust planning and execution systems for virtual worlds. In: *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE)*, pp. 29–35 (2006)
13. Michalewicz, Z., Nazhiyath, G.: Genocop III: A co-evolutionary algorithm for numerical optimization with nonlinear constraints. In: Fogel, D.B. (ed.) *Proceedings of the Second IEEE International Conference on Evolutionary Computation*, pp. 647–651. IEEE Press, Piscataway (1995)
14. Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D.: Displacement problem and dynamically scheduling aircraft landings. *Journal of the Operational Research Society* 55, 54–65 (2004)
15. Gao, J., Sheng, Z.: Research for dynamic vehicle routing problem with time windows in real city environment. In: *Proceedings of the 2008 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, Piscataway, NJ, USA, vol. 2, pp. 3052–3056 (2008)
16. Gleicher, M., Ferrier, N.: Evaluating Video-Based Motion Capture. In: *CA 2002: Proceedings of the Computer Animation*, pp. 75–80. IEEE Computer Society, Washington, DC (2002)
17. Daugulis, A.J., McLellan, P.J., Li, J.: Experimental investigation and modeling of oscillatory behavior in the continuous culture of *Zymomonas mobilis*. *Biotechnology and Bioengineering* 56(1), 99–105 (1997)
18. Haugwitz, S., Hagander, P., Norn, T.: Modeling and control of a novel heat exchange reactor, the Open Plate Reactor. *Control Engineering Practice* 15(7), 779–792 (2007)
19. Angeline, P.J.: Tracking extrema in dynamic environments. In: Angeline, P.J., McDonnell, J.R., Reynolds, R.G., Eberhart, R. (eds.) *EP 1997. LNCS*, vol. 1213, pp. 335–345. Springer, Heidelberg (1997)
20. Bird, S., Li, X.: Informative performance metrics for dynamic optimisation problems. In: *GECCO 2007: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pp. 18–25. ACM, New York (2007)
21. Branke, J., Salihoglu, E., Uyar, S.: Towards an Analysis of Dynamic Environments. In: Beyer, H.G., et al. (eds.) *Genetic and Evolutionary Computation Conference*, pp. 1433–1439. ACM (2005)
22. Isaacs, A., Puttige, V.R., Ray, T., Smith, W., Anavatti, S.G.: Development of a memetic algorithm for Dynamic Multi-Objective Optimization and its applications for online neural network modeling of UAVs. In: *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008*, pp. 548–554. IEEE (2008)
23. Tawdross, P., Lakshmanan, S.K., Konig, A.: Intrinsic Evolution of Predictable Behavior Evolvable Hardware in Dynamic Environment. In: *HIS 2006: Proceedings of the Sixth International Conference on Hybrid Intelligent Systems*, p. 60. IEEE Computer Society (2006)
24. Rocha, M., Neves, J., Veloso, A.: Evolutionary Algorithms for Static and Dynamic Optimization of Fed-batch Fermentation Processes. In: Ribeiro, B., et al. (eds.) *Adaptive and Natural Computing Algorithms*, Springer (2005)

25. Mertens, K., Holvoet, T., Berbers, Y.: The DynCOAA algorithm for dynamic constraint optimization problems. In: AAMAS 2006: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1421–1423. ACM, New York (2006)
26. Ioannou, P., Chassiakos, A., Jula, H., Unglaub, R.: Dynamic optimization of cargo movement by trucks in metropolitan areas with adjacent ports. Technical report, METTRANS Transportation Center, University of Southern California, Los Angeles, CA 90089, USA (2002)
27. Andrews, M., Tuson, A.L.: Dynamic Optimisation: A Practitioner Requirements Study. In: Proceedings of the The 24th Annual Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG 2005), London, UK (2005)
28. Schlegel, M., Marquardt, W.: Adaptive switching structure detection for the solution of Dynamic Optimization Problems. *Industrial & Engineering Chemistry Research* 45(24), 8083–8094 (2006)
29. Wang, Y., Wineberg, M.: Estimation of evolvability genetic algorithm and dynamic environments. *Genetic Programming and Evolvable Machines* 7(4), 355–382 (2006)
30. Prata, D.M., Lima, E.L., Pinto, J.C.: Simultaneous Data Reconciliation and Parameter Estimation in Bulk Polypropylene Polymerizations in Real Time. *Macromolecular Symposia* 243(1), 91–103 (2006)
31. Padula, S., Gumbert, C., Li, W.: Aerospace applications of optimization under uncertainty. *Optimization and Engineering* 7(3), 317–328 (2006)
32. Araujo, L., Merelo, J.J.: A genetic algorithm for dynamic modelling and prediction of activity in document streams. In: GECCO 2007: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, pp. 1896–1903. ACM, New York (2007)
33. Deb, K., Rao Udaya Bhaskara, N., Karthik, S.: Dynamic Multi-objective Optimization and Decision-Making Using Modified NSGA-II: A Case Study on Hydrothermal Power Scheduling. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 803–817. Springer, Heidelberg (2007)
34. 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)
35. Mezura-Montes, E., Coello Coello, C.A.: Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation* 1(4), 173–194 (2011)

Adaptive Differential Evolution with Difference Mean Based Perturbation for Practical Engineering Optimization Problems

Rupam Kundu¹, Rohan Mukherjee¹, and Swagatam Das²

¹ Electronics and Telecommunication Engineering Department, Jadavpur University

² Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata
{rupam2422,rohan.mukherjee}@gmail.com, swagatam.das@isical.ac.in

Abstract. Differential Evolution(DE) is one of the most versatile evolutionary techniques that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Recent developments on DE includes self adaptation of its parameters (F =step size and CR =cross-over probability) making it a parameter free optimizer. A new self adaptive DE(jDE) proposed by Janez Brest, is a robust improvement of DE, where the self adaptive parameters undergo similar operations of genetic operators. This paper aims at introducing a unique mutation strategy by modifying the existing "DE/rand/1/bin" strategy of jDE with Difference Mean Based Perturbation (DMP) technique. The algorithm addressed as ADE-DMP is basically a variant of jDE, but the modified mutation scheme ensues within the algorithm effective search of area near the current best that effectively proves it to be a better and fast optimizer in complex real world problems of diverse domains.

1 Introduction

Differential Evolution(DE) is arguably one of the most effective evolutionary techniques of global optimization known for its simplicity, fast convergence and its manifold applications in various field of optimization including scientific and engineering fields, introduced by Storn and Price[1–4]. Proper tuning of the parameters is an integral part for realizing quality solutions by DE. The common practiced method includes the selection of a value for a particular parameter and then tuning it by repeated application keeping it to be fixed throughout the run. But the choice of parameters for a particular problem may not be the correct choice for another problem. Parameter control essentially brought the revolution for DE that eventually proves it to be one of the best optimization techniques of the present era. Recent works include self adaptive parameter control where the parameters to be adjusted are encoded in the individuals and undergo the similar operations of genetic operators. Such an improvement on a basic DE framework can be observed in a new self adaptive DE, jDE by Janez Brest[5].

In this paper the mutation scheme used in jDE, "DE/rand/1/bin" has been modified with a unique upgradation technique called Difference Mean based Perturbation(DMP) that offers a perfect blend of exploration and exploitation to the

original version of jDE. The improvement of the mutation scheme over the one used in jDE is clearly reflected in the performance of ADE-DMP on comparison with other state-of-artEAs proposed in literature as well as jDE itself. The stable yet steady rate of convergence offered by DMP mutation is also evident from the convergence characteristics of our method. Some selected problems of the real world optimization problems of Competition on Testing Evolutionary Algorithms on Real-world Numerical Optimization problems, held under the 2011 *IEEE Congress on Evolutionary Computation(CEC 2011)*[6] are chosen to judge the performance of ADE-DMP. The problems are quite diverse in the respect that they include low and high dimensionality problems with unequal bound constraints, high complexity and constraint in the function evaluations. Function Evaluation is supposed to be the biggest challenge for all the EAs in the competition as minimizing a high dimensionality problem within such a limited function evaluations, to obtain the best optimal combinations for the particular problem requires a powerful optimization technique that is both fast and efficient. The proposed method successfully outperforms the other approaches in the competition satisfying the constraint of function evaluations in a statistically significant way, thereby yielding the best optimal solutions in most of the test cases considered.

2 Adaptive DE with Difference Mean Based Perturbation (ADE-DMP)

The convergence rate of jDE is no doubt better than the conventional DE algorithm but when applied on the real world problems as mentioned in [6] jDE fails to obtain the best optimal solution within limited function evaluations. The algorithm remains slow to convergence speed required and fails to attain optimal solution. The problems with high dimensionality and high complexity necessarily demand an approach that sufficiently provides an optimal tradeoff between exploration and exploitation so that desired solution can be obtained within minimum function evaluations. The proposed method modifies the mutation scheme for jDE to satisfy the above said demand thereby making jDE fast in terms of convergence and also by embedding an explorative characteristic to the conventional jDE search mechanism. The mutation scheme is updated as follows:

$$v_i^{G+1} = x_{r1}^G + F_i^{G+1}[(x_{r2}^G - x_{r3}^G) + (\omega * \textit{directed_vector}_i)], \quad (1)$$

where $r1, r2, r3$ are randomly chosen indices from $[1, NP]$ where NP is the population size, and ω is the inertia factor. v_i^{G+1} denotes the mutation vector for i -th individual in $G+1$ -th generation and x_{r1}^G denotes a randomly chosen individual in G -th generation. Generation of *directed_vector* is discussed in Section 2.1. Here the F and CR parameters are evolved similar to the procedure as described in jDE [5].

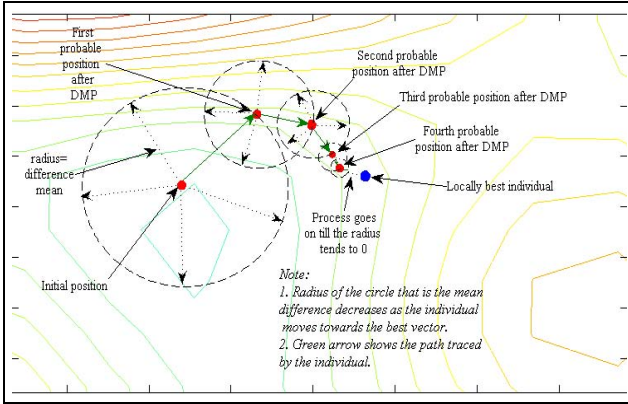


Fig. 1. Perturbation with difference mean based scaling coefficient-an illustration

2.1 Directed_Vector Generation

The *directed_vector_i* is generated according to a Difference Mean based Perturbation (DMP) scheme which is explained in Table 1.

Outline of DMP. The DMP scheme amounts to simply perturbing (mutating) a newly generated population member with a scaled unit vector along any random direction. The unit vector is scaled by the difference mean formed by subtracting the dimensional mean of the individual to be perturbed from the current best individual of the population. Note that dimensional mean is a scalar quantity obtained by averaging the components of a vector individual from the population. As will be discussed subsequently, DMP causes an individual to search effectively near current best.

Motivation behind DMP Scheme. In DMP the mean of the components of the best vector is subtracted from the corresponding mean of the target individual selected for perturbation. The result is a difference mean which is used to scale a unit vector in random direction. This ensues within the scheme, a mix of explorative and exploitative behavior, retaining the true essence of attraction towards the best individual. Suppose we consider two vectors \mathbf{A} and \mathbf{B} , where \mathbf{A} is the best vector and \mathbf{B} can be any random individual from the population. Mean can be chosen as a measure of the central tendency as average of a set of values is highly dependent on the outlying observation that appears to deviate markedly from other members of the sample in which it occurs, known as outliers[7]. The arithmetic mean indicates the central tendency of a collection of numbers taken as the sum of the numbers divided by the size of the collection. Thus, if we consider the dimensional values of \mathbf{A} to form a set say $a = \{a_1, a_2, \dots, a_n\}$ and that of \mathbf{B} as $b = \{b_1, b_2, \dots, b_n\}$, then both a and b has a definite mean each. Let us denote these with \bar{a} and \bar{b} . Let there exist at least one $b_i, i \in [1, n]$ such that b_i is greater or smaller than a_i by a significant margin that

largely contributes to the difference between the best vector \mathbf{A} and any vector \mathbf{B} . Suppose the other component of vectors \mathbf{A} and \mathbf{B} are more or less close to each other. As per the above assumptions, the mean of b will be highly influenced by b_i and the difference between \bar{a} and \bar{b} will be proportional to b_i . There may be positive or negative deviation from the central mean i.e. the mean of the best individual. Therefore difference mean can very well determine the spread of the population members. It increases with the spread of solutions surrounding the best individual, but it tends to zero with convergence. Instead of random scaling while perturbing an individual, the difference mean can be used as an effective scaling coefficient. The unit random vector generated from a normal distribution, adds exploration to the scheme as the direction and sense of the perturbation can be random. Fig. 1 shows the gradual decrease in the difference mean as the individual moves towards the locally best individual using DMP. If an individual is perturbed only by DMP scheme then the update equation will be:

$$x_{i,j} = x_{i,j} + \text{directed_vector}_{i,j}. \quad (2)$$

$\forall i = (k_1, k_2, \dots, NP)$ and $\forall j = (1, 2, \dots, D)$. As we can see in Fig. 1, initially the individual being far away from the best individual generates a large difference mean, and hence the sphere within which new solution is to be generated by eqn.(3), has comparatively large radius. Now using only DMP repetitively it can be shown that as the mean difference between the best vector and the other individual is reduced, the radius of the sphere tends to zero and the individual gradually converge. It is to be noted that since we have considered a contour plot, we have used circle in the figure which is a projection of sphere in horizontal plane. It is evident from the figure that this update scheme ensures an effective search of the area near the current best. The arrows shown within the circle signify the possible movements of the individual in those directions indicating the explorative nature of the scheme. We can thus conclude that the difference mean based perturbation scheme retains the influence of the attraction towards the best individual by reducing the difference mean providing a scope for convergence and at the same time the randomly directed unit vector enhances the explorative characteristics.

Problems of DE Variants in High Dimensionality and Possible Remedy by DMP. DE variants are characterized by localized search around part of the search space for better exploitation of the source. The parents taking part in mutation and recombination produce offsprings and guide them in favorable parts of the search space which leads to better convergence in search spaces. Unlike previous EAs where search mechanism of the parent population was not generally restricted to a particular direction, DE evolved with this enhanced optimizing technique. We explain the above fact with the following example. DE mutation in 1- D (where D is the dimension of the problem) field with two particles will produce solution only in that part in which the resultant vector will get directed. In two dimensional fields this solution will get concentrated to one of the four possible quadrants. Usage of CR parameter in DE aids optimization by aligning part of the vector solution towards one of the parent which already

enjoyed superior place in the landscape. In other words the vector obtained by mutation is rotated towards favorable parts leading to an improved EA. This makes DE unique and superior to PSO. The concentrated local search in parts of the landscape that is guided by parent vectors makes DE exploitation much effective than Genetic Algorithm[8]. However concentrated search of DE with increasing dimensionality is not always suitable. In one dimension the search is in 1/2 part of the landscape, the possible directions being left or right. In 2- D space, the search is in 1 out of 4 quadrants and thus 1/4 part of the landscape gets explored. In 3- D space, the search is in 1/8 of the search space and so on. Thus with increasing dimensionality the search gets more and more concentrated. As time devoted to optimization in higher dimensionality spaces is quite adequate this does not provide serious hindrance to optimization till a specific level. However if dimension is soared as high as 1000 or more as seldom is the case for high dimensional optimization or practical life problems performance of DE as well as some DE variants that involve DE alike steps is hampered. The reason is quite obvious from previous analogies. With dimensionality the search space is decreasing as $1/(2^D)$. Naturally as D increases the search region diminishes to very little portion of the Hyper-sphere. A very simple remedy to such problem is generating solutions all through the landscape as used by algorithms like GA. However such an optimization technique by random movement in random direction may not be very reliable. What we require is a practical optimized scenario by minimizing the trade off error between the two possible techniques, one for guided exploitation like DE and other for added exploration like GA. Retaining the essence of mutation and recombination of classical DE, DMP overcomes the exhaustive search technique of DE by the use of randomly generated directed vector besides incorporating the explorative characteristics of GA. The direction defined in a mutation scheme in classical DE is strictly determined by the difference between two vectors and is certainly biased by the positional information of the two selected vectors. But in DMP the directional vector defined by DE is approximated by its dimensional mean since we have already discussed earlier that the mean can be a very good statistical measure of a vectors positional information. The dimensional mean is thus used to scale a randomly directed vector in search space which ensures uniform initialization of offspring vector. The randomization incorporated in the jDE mutation in eqn. (1) thus essentially provides an optimal trade off between the concentrated and exhaustive search technique offered by DE besides ensuring a uniform offspring generation over the entire search domain. This feature can be declared as the most important aspect of ADE-DMP as the mutation used here gains pre-dominance especially in high dimensional scenarios, which justifies its performance in the test problems considered in this paper.

2.2 Effect on Successful Updates per Generation

The modification on the mutation scheme can be realized with the increase in successful updates per generation compared to jDE. If the new set of control parameters can produce offsprings that eventually survives under DE selection,

then that individual may be declared as a successful update. The increased successful updates have a direct influence on the fast convergence of the algorithm. It is evident from Fig. 2(b) that the average successful update at any point of the search procedure is always greater than that of jDE. Moreover Fig. 2(a) shows that the function evaluations required to obtain the same objective functional value by ADE-DMP and jDE are much less in case of ADE-DMP thereby establishing the fact that it is a quite faster approach than jDE.

Table 1: Algorithm: *directed_vector* Generation

Step1. Calculate the dimensional mean of the best individual of a population:

$$best_avg = \frac{1}{D} \sum_{j=1}^D (x_{best,j}), \tag{3}$$

where D is the number of search space dimensions and indicates the j th component of the current best population member \mathbf{x}_{best} .

Step2. Now calculate the dimensional mean of the i th member of the population:

$$ind_avg_i = \frac{1}{D} \sum_{j=1}^D (x_{i,j}), \forall i = (k_1, k_2, \dots, NP). \tag{4}$$

Step3. Calculate the difference between the mean of the best and mean of the i th member of the population obtained in *Step2*..

$$diff_mean_i = best_avg - ind_avg_i. \tag{5}$$

Step4. Generate a vector with components chosen randomly from unit normal distribution with zero mean and unity variance:

$$\mathbf{v} = \{v_1, v_2, \dots, v_d\}, v_i \in N(0, 1), \forall i \in (1, 2, \dots, D). \tag{6}$$

Step5. The *directed_vector* is thus generated by scaling \mathbf{v} by the obtained difference mean for the corresponding individual:

$$directed_vector_i = diff_mean_i * \hat{v}, \forall i = (k_1, k_2, \dots, NP), \tag{7}$$

where, $\hat{v} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$.

2.3 Effect of Inertia Weight

The use of inertia factor ω in eqn.(2) controls the influence of directed vector in the mutation scheme. The DMP mutation imparts into the mutation scheme a balanced combination of exploration and exploitation. But the use of $directed_vector_i$ till the end may cause the population to converge prematurely. So to avoid early stagnation the effect of $directed_vector_i$ i.e. the term $F_i^{G+1} * directed_vector_i$ in eqn.(1) has been reduced gradually during the later stage of the search procedure to maintain required spread in the population. The inertia factor ω is given as:

$$\omega = 0.9 - 0.5 * (1 - \frac{1}{G}), \quad G = \text{current generation.} \quad (8)$$

Table 2 justifies the necessity of ω in eqn. (2) by tabulating the results obtained on selected problems [TP10, TP11.1, TP11.6, T11.9] of [6] using ω as mentioned in eqn.(1) and without using ω in eqn.(1) i.e. replacing ω by 1. It is evident from Table 2 that the control of DMP mutation by use of inertia weight highly influences the performance of ADE-DMP.

Table 2. Comparison between the results obtained by ADE-DMP using and without using inertia weight

| Test Problem | Without ω | With ω |
|--------------|------------------|-----------------|
| TP10 | -21.45 | -21.74 |
| T11.1 | 5.19E+04 | 5.14E+04 |
| TP11.6 | 1.22E+05 | 1.21E+05 |
| TP11.9 | 9.26E+06 | 9.25E+06 |

3 Results and Discussions

ADE-DMP is tested using a set of selected benchmark functions (T10, T11.1-T11.10) from the Competition on Testing Evolutionary Algorithms on Real-world

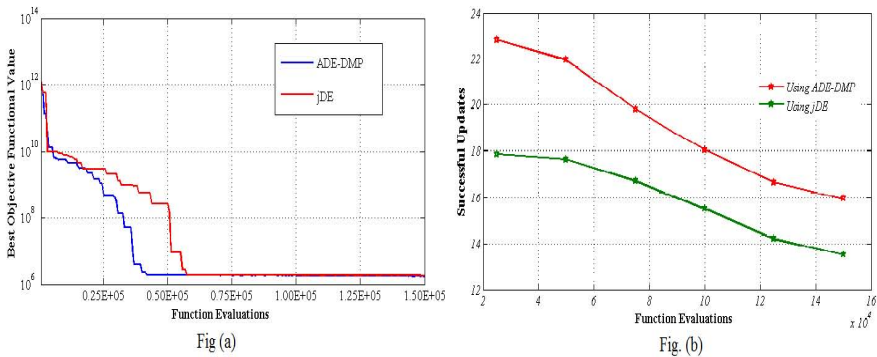


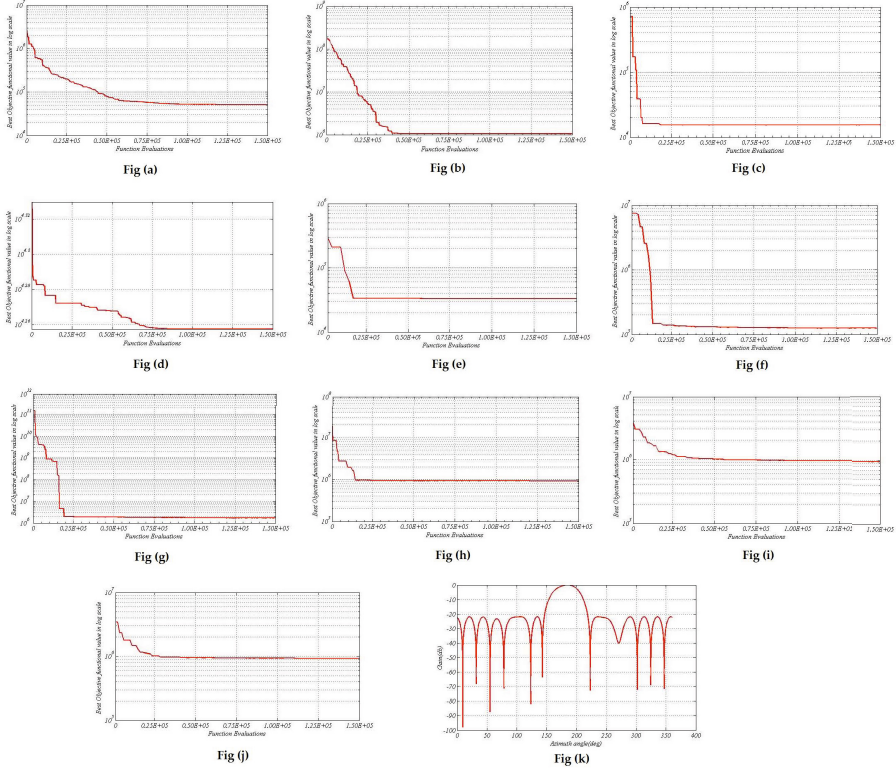
Fig. 2. (a) Successful Update vs. Function Evaluations using ADE-DMP mutation and jDE mutation (b) Convergence characteristic for TP11.7

Table 3. Best objective functional values achieved by ADE-DMP and other well-known state-of-artEAs

| Function | | TP10 | TP11.1 | TP11.2 | TP11.3 | TP11.4 | TP11.5 | TP11.6 | TP11.7 | TP11.8 | TP11.9 | TP11.10 | Average Rank |
|---------------|---------|----------|----------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|--------------|
| GA-MPC | Average | -21.64 | 4.94E+04 | 1.06E+06 | 1.54E+04 | 1.81E+04 | 3.27E+04 | 1.22E+05 | 1.78E+06 | 9.25E+05 | 9.31E+05 | 9.24E+05 | 2.09 |
| | Std | 1.16E-01 | 1.16E+03 | 2.59E+03 | 1.75E-07 | 4.87E+01 | 6.96E+00 | 1.55E+02 | 6.84E+04 | 8.14E+02 | 1.98E+03 | 1.29E+02 | |
| SAMODE | Average | -21.6 | 5.33E+04 | 1.07E+06 | 1.54E+04 | 1.82E+04 | 3.27E+04 | 1.24E+05 | 1.99E+06 | 9.26E+05 | 9.26E+05 | 9.26E+05 | 3.91 |
| | Std | 1.13E-01 | 2.03E+03 | 1.27E+03 | 3.73E-02 | 1.07E+02 | 7.77E+00 | 8.07E+02 | 1.26E+05 | 6.37E+02 | 7.29E+02 | 7.61E+02 | |
| CDASA [23] | Average | -14.29 | 5.20E+04 | 1.27E+06 | 1.55E+04 | 1.93E+04 | 3.32E+04 | 1.47E+05 | 2.04E+06 | 9.46E+05 | 1.40E+06 | 9.49E+05 | 8.09 |
| | Std | 1.89E+00 | 3.90E+02 | 1.87E+03 | 4.23E+01 | 2.13E+02 | 1.05E+02 | 6.34E+03 | 2.41E+05 | 3.69E+03 | 1.79E+05 | 1.81E+04 | |
| ED.DE | Average | -21.15 | 5.21E+04 | 1.08E+06 | 1.54E+04 | 1.83E+04 | 3.29E+04 | 1.33E+05 | 1.91E+06 | 9.41E+05 | 1.10E+06 | 9.41E+05 | 5.09 |
| | Std | 7.69E-01 | 3.66E+02 | 2.73E+04 | 9.10E+00 | 8.70E+01 | 5.95E+01 | 1.94E+03 | 9.10E+03 | 1.60E+03 | 6.43E+04 | 1.60E+03 | |
| EA.DE.MA [24] | Average | -21.34 | 1.52E+06 | 2.06E+07 | 1.55E+04 | 1.92E+04 | 3.29E+04 | 3.52E+05 | 2.58E+06 | 1.66E+06 | 2.35E+06 | 1.66E+06 | 9.55 |
| | Std | 9.04E-01 | 7.18E+05 | 6.06E+05 | 4.24E+00 | 1.24E+02 | 4.92E+01 | 6.09E+03 | 6.47E+05 | 3.28E+05 | 4.55E+05 | 3.28E+05 | |
| DE-RHC [25] | Average | -18.52 | 5.31E+04 | 1.81E+07 | 1.55E+04 | 1.92E+04 | 3.31E+04 | 1.39E+05 | 2.06E+06 | 1.09E+06 | 1.50E+06 | 1.05E+06 | 8.91 |
| | Std | 1.19E+00 | 5.11E+02 | 4.40E+04 | 8.78E+00 | 1.54E+02 | 7.46E+01 | 3.19E+03 | 2.38E+05 | 1.56E+05 | 2.35E+05 | 1.39E+05 | |
| ENSMI-DE [14] | Average | -15.2 | 5.24E+04 | 1.07E+06 | 1.55E+04 | 1.82E+04 | 3.28E+04 | 1.31E+05 | 1.92E+06 | 9.43E+05 | 9.90E+05 | 9.43E+05 | 5.27 |
| | Std | 3.79E+00 | 7.01E+02 | 2.03E+03 | 1.55E+01 | 4.19E+01 | 3.73E+01 | 2.47E+03 | 1.18E+04 | 2.63E+03 | 4.14E+04 | 2.63E+03 | |
| MOD-DE-LS | Average | -20.78 | 7.38E+04 | 2.44E+06 | 1.54E+04 | 1.91E+04 | 3.30E+04 | 1.38E+05 | 1.95E+06 | 1.02E+06 | 1.54E+06 | 1.03E+06 | 7.64 |
| | Std | 6.50E-01 | 8833.496 | 76960.21 | 1.666127 | 90.05969 | 53.00626 | 2682.525 | 18708.56 | 159589.2 | 219368.6 | 150437.6 | |
| WI-DE [26] | Average | -21.54 | 5.41E+04 | 1.08E+06 | 1.54E+04 | 1.82E+04 | 3.28E+04 | 1.36E+05 | 2.06E+06 | 1.06E+06 | 1.14E+06 | 1.08E+06 | 6.14 |
| | Std | 1.50E-01 | 4.72E+03 | 9.04E+03 | 0.00E+00 | 5.45E+01 | 4.23E+01 | 2.57E+03 | 1.58E+05 | 7.53E+04 | 1.49E+05 | 7.04E+04 | |
| RGA [27] | Average | -17.35 | 5.31E+04 | 1.78E+07 | 1.55E+04 | 1.91E+04 | 3.29E+04 | 3.52E+05 | 2.09E+06 | 9.57E+05 | 1.44E+06 | 9.57E+05 | 8.18 |
| | Std | 6.18E+00 | 7.87E+02 | 76960.21 | 1.666127 | 90.05969 | 53.00626 | 2682.525 | 18708.56 | 159589.2 | 219368.6 | 150437.6 | |
| ADE-DMP | Average | -21.74 | 5.14E+04 | 1.05E+06 | 1.54E+04 | 1.80E+04 | 3.26E+04 | 1.21E+05 | 1.66E+06 | 9.23E+05 | 9.25E+05 | 9.23E+05 | 1.13 |
| | Std | 1.13E-01 | 3.87E+02 | 2.56E+03 | 0E+00 | 2.83E+01 | 4.04E+00 | 1.74E+01 | 2.12E+02 | 1.88E+03 | 8.62E+02 | 1.88E+03 | |

Numerical Optimization problems, held under the 2011 *IEEE Congress on Evolutionary Computation(CEC 2011)*[6]. These functions span a diverse set of problem features, including multimodality, high dimensionality and high complexity. The mean best-of-the-run errors and standard deviations over 50 independent runs are reported for 11 functions in Table 3. Each run on each function is continued till a maximum number of Function Evaluations (FEs) equal to 1.5E+05 as recommended in [6]. The population size NP is kept fixed at 100 for all the functions. All the runs have been taken on a machine with Windows 7 64-bit configuration, 4GB RAM, 500GB hard disk and 2.26 GHz Intel Core i3-350M Processor. Wilcoxon rank sum test for independent samples [9] is conducted at the 5% significance level in order to judge whether the results obtained with the best

Sample Convergence graphs obtained by ADE-DMP on all test problems



Convergence graph for Fig.(a) TP11.1 Fig.(b) TP11.2 Fig.(c) TP11.3 Fig.(d) TP11.4 Fig.(e) TP11.5 Fig.(f) TP11.6 Fig.(g) TP11.7 Fig.(h) TP11.8 Fig.(i) TP11.9 Fig.(j) TP11.10 Fig.(k) TP10

performing algorithm differ from the final results of rest of the competitors in a statistically significant way. P-values obtained through the rank sum test between the best algorithm and each of the remaining algorithms over all the functions are presented in Table 3. For the P-value entries NA stands for Not Applicable and occurs for the best performing algorithm itself in each case. If the P-values are less than 0.05 (5% significance level), it is a strong evidence against the null hypothesis, indicating that the better final objective function values achieved by the best algorithm in each case is statistically significant and has not occurred by chance. Analysis of Table 3 reveals that ADE-DMP consistently outperforms GA-MPC[10] and other DE variants in 10 out of 11 cases in a statistically significant way. In test case T11.1 GA-MPC performs marginally better than ADE-DMP. In the Circular Antenna Array Design Problem (TP10) also ADE-DMP is successful in obtaining the best optimal combination of current and phase excitations. Ranking has been done for each algorithm on the basis of their performance on each function and average rank has been reported.

When more than one algorithm has the same result ranking has been done on the basis of the standard deviation. Considering the above assumption WI-DE and ADE-DMP are ranked better than the other algorithms in case of TP11.3 though GA-MPC, SAMODE[11], ED-DE[12], MOD-DE-LS[13] has obtained the same result.

To study the convergence characteristics of ADE-DMP, the Best Objective Functional Value in log scale vs. Function Evaluations is plotted for all the cases T11.1-T11.10 mentioned above. In case of T10 radiation pattern(Gain(in db) vs Azimuthal Angle) for circular antenna array in optimal setting is plotted.

4 Conclusion

In this paper the effect of modifying the mutation scheme of jDE by addition of directed vector generated by DMP technique has been explored and different test conditions are used in this context to verify our proposals. The DMP strategy involves filtering out the best aspect of every dimension associated with a functionally favourable individual and using it to guide any other individual towards the global optimum. The best aspect as discussed is conveyed by the mean of all the best aspects and hence the use of the mean based up gradation. The mean difference is used to scale a randomly directed vector which when added to the selected individual to be perturbed is guided through random paths towards the global best. Thus a balanced exploration and exploitation is readily satisfied. In addition, DMP scheme is free from any parameters. The use of mutation factor for each directed vector increases the robustness of the scheme. Moreover the controlled use of the directed vector prevents premature stagnation and provides a steady rate of convergence. Also it has been shown that ADE-DMP has a faster rate of convergence than conventional jDE which is an essential criterion for optimization with constrained function evaluations. In spite of the fact that the problems chosen to test the efficacy of ADE-DMP offers high dimensionality, high complexity, multimodality etc. the proposed method has proved its superiority over most of the other renowned approaches thereby establishing it as a fast and promising optimizer.

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. Storn, R., Price, K.V., Lampinen, J.: *Differential Evolution - A Practical Approach to Global Optimization*. Springer, Berlin (2005)
3. 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)
4. Islam, S.M., Das, S., Ghosh, S., Roy, S., Suganthan, P.N.: An Adaptive Differential Evolution Algorithm With Novel Mutation and Crossover Strategies for Global Numerical Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 42(2), 482–500 (2012)

5. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. *IEEE Transactions on Evolutionary Computation* 10(6) (December 2006)
6. Das, S., Suganthan, P.N.: Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems. Technical Report (December 2010)
7. Strutz, T.: *Data Fitting and Uncertainty - "A practical introduction to weighted least squares and beyond"*. Vieweg+Teubner (2010)
8. 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)
9. Derrac, J., Garca, 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)
10. Elsayed, S., Sarker, R., Essam, D.: GA with a New Multi-Parent Crossover for Solving IEEE-CEC 2011 Competition Problems. In: *2011 IEEE Congress on Evolutionary Computation (CEC)*, New Orleans, pp. 1034–1040 (June 2011)
11. Elsayed, S., Sarker, R., Essam, D.: Differential Evolution with Multiple Strategies for Solving CEC2011 Real-world Numerical Optimization Problems. In: *2011 IEEE Congress on Evolutionary Computation (CEC)*, New Orleans, pp. 1041–1048 (June 2011)
12. Wang, Y., Li, B., Zhang, K.: Estimation of Distribution and Differential Evolution Cooperation for Real-world Numerical Optimization Problems. In: *2011 IEEE Congress on Evolutionary Computation (CEC)*, June 5–8, pp. 1315–1321 (2011)
13. Mandal, A., Das, A.K., Mukherjee, P., Das, S., Suganthan, P.N.: Modified differential evolution with local search algorithm for real world optimization. In: *2011 IEEE Congress on Evolutionary Computation (CEC)*, June 5–8, pp. 1565–1572 (2011)
14. Korosec, P., Silc, J.: The Continuous Differential Ant-Stigmergy Algorithm Applied to Real-World Optimization Problems. In: *2011 IEEE Congress on Evolutionary Computation (CEC)*, New Orleans, pp. 1327–1334 (June 2011)
15. Singh, H.K., Ray, T.: Performance of a Hybrid EA-DE-Memetic Algorithm on CEC 2011 Real World Optimization Problems. In: *2011 IEEE Congress on Evolutionary Computation (CEC)*, New Orleans, pp. 1322–1326 (June 2011)
16. LaTorre, A., Muelas, S., Pena, J.: Benchmarking a Hybrid DE-RHC Algorithm on Real World Problems. In: *2011 IEEE Congress on Evolutionary Computation (CEC)*, New Orleans, pp. 1027–1033 (June 2011)
17. Haider, U., Das, S., Maity, D., Abraham, A., Dasgupta, P.: Self Adaptive Cluster Based and Weed Inspired Differential Evolution Algorithm For Real World Optimization. In: *2011 IEEE Congress on Evolutionary Computation (CEC)*, New Orleans, pp. 750–756 (June 2011)
18. Saha, A., Ray, T.: How does the good old Genetic Algorithm fare at Real World Optimization? In: *2011 IEEE Congress on Evolutionary Computation (CEC)*, New Orleans, pp. 1049–1056 (June 2011)

Transmission Line Management Using Multi-objective Evolutionary Algorithm

K. Pandiarajan¹ and C.K. Babula²

¹ Department of Electrical and Electronics Engineering,
Pandian Saraswathi Yadav Engineering College, Arasanoor, India
P_raajan@rediffmail.com

² Department of Electrical and Electronics Engineering,
Thiagarajar College of Engineering, Madurai, India
ckbeee@tce.edu

Abstract. This paper presents an effective method of transmission line management in power systems. Two conflicting objectives 1) generation cost and 2) transmission line overload are optimized to provide non-dominated Pareto-optimal solutions. A fuzzy ranking-based multi-objective differential evolution (MODE) is used to solve this complex nonlinear optimization problem. The generator real power and generator bus voltage magnitude is taken as control variables to minimize the conflicting objectives. The fuzzy ranking method is employed to extract the best compromise solution out of the available non-dominated solutions depending upon its highest rank. N-1 contingency analysis is carried out to identify the most severe lines and those lines are selected for outage. The effectiveness of the proposed method has been analyzed on standard IEEE 30 bus system with smooth cost functions and their results are compared with non-dominated sorting genetic algorithm-II (NSGA-II) and Differential evolution (DE). The results demonstrate the superiority of the MODE as a promising multi-objective evolutionary algorithm to solve the power system multi-objective optimization problem.

1 Introduction

Optimal power flow (OPF) is an important tool for power system management. The aim of OPF problem is to optimize one or more objectives by adjusting the power system control variables while satisfying a set of physical and operating constraints such as generation and load balance, bus voltage limits, power flow equations, and active and reactive power limits. A variety of optimization techniques had been applied to solve the OPF problem such as gradient method [1], linear programming method [2] and interior point method. In conventional optimization methods, identification of global minimum is not possible. To overcome the difficulty, evolutionary algorithms like genetic Algorithm (GA) [3], particle swarm optimization [4], differential evolution [5], gravitational search algorithm [6], tabu search algorithm [7] and artificial bee colony algorithm [8] had been proposed.

In [9], the authors' proposed a fuzzy logic based approach to alleviate the network overloads by generation rescheduling. The generation shift sensitivity factor (GSSF)

was used to decide the changes in generation. In [10], the authors' proposed an optimal location of interline power flow controller (IPFC) in a power system network using artificial bee colony algorithm (ABC). Minimization of line loss, economic dispatch of generators, improve power flow and reduction in the overall system cost which includes the cost of active power generation and the installation cost of IPFC were also considered for obtaining the optimal location. In [11], the authors' proposed a static security enhancement through optimal utilization of thyristor-controlled series capacitors (TCSC). The branches ranking in the system was based on determination of single contingency sensitivity (SCS) index which helped to decide the best locations for the TCSCs. The objective of the optimization problem was to eliminate or minimize line overloads as well as the unwanted loop flows under single contingencies. In [12], the authors' proposed the use of genetic algorithm (GA) and multi-objective genetic algorithm (MOGA) to alleviate the violations of the overloaded lines and minimize the transmission power losses for different operating conditions. In [13], the authors' proposed a multi-objective particle swarm optimization (MOPSO) method for transmission line overload management. Two competing objectives were considered for minimization such as line overloads and operating cost of generators. The overloads in a transmission network were alleviated by generation rescheduling. In [14], the authors' proposed a graphical user interface (GUI) based on a genetic algorithm. It was used to determine the optimal location and sizing parameters of multi type FACTS devices which facilitate maximization of power system loadability in a transmission network. In [15], the authors' proposed a non-dominated sorting genetic algorithm (NSGA), niched multi-objective genetic algorithm (NPGA) and strength multi-objective evolutionary algorithm (SPEA) to minimize two competing objective functions such as fuel cost and emission. The results of these proposed methods were compared to each other. The SPEA method had better diversity characteristics and was more efficient when compared to other methods. In [16], the authors' proposed an application of hybrid differential evolution with particle swarm optimization (DEPSO) to solve the maximum loadability problem. The results were compared with multi agent hybrid particle swarm optimization (MAHPSO) and differential evolution (DE). This proposed algorithm had improved the loadability margin with less number of iterations by consuming more time per iteration when compared to other algorithms. In [17], the authors' proposed a survey on development of multi-objective evolutionary algorithms (MOEAs). It covered algorithmic frameworks such as decomposition-based MOEAs (MOEA/Ds), memetic MOEAs, co-evolutionary MOEAs, selection and offspring reproduction operators, MOEAs with specific search methods, MOEAs for multimodal problems, constraint handling and MOEAs, computationally expensive multi-objective optimization problems (MOPs), dynamic MOPs, noisy MOPs, combinatorial and discrete MOPs, benchmark problems, performance indicators and applications.

In this paper, a fuzzy ranking based multi-objective differential evolution for overload management in power system network is presented with an illustrated example.

The organization of the paper is as follows: Section 2 presents the optimization problem formulation for transmission line overload management. Section 3 presents the algorithm of proposed MODE for transmission line overload management. The simulation results for different contingency cases in IEEE 30 bus system is presented in section 4. Finally, conclusion and future works is given in Section 5.

2 Problem Formulation

The objective function of the proposed method is to find an optimum value of shift in active power generation and generator bus voltage magnitude along with network constraints so as to minimize the total generation cost and line overloads simultaneously in the network. The problem of proposed MODE may be stated as follows.

2.1 Objective Function

Objective 1. Minimize total generation cost

$$\text{Generation cost, } GC = \sum_{i=1}^{NG} (a_i P_{gi}^2 + b_i P_{gi} + C_i) \quad (1)$$

where:

- GC Generation cost
- N_G Number of participating generators
- P_{gi} Generation of i^{th} generator
- a_i, b_i, c_i Cost coefficients of generator i

Objective 2. Minimize transmission line overload by reducing Overload Index

$$\text{Overload Index, } OI = \sum_{i=1}^{N_L} (LF_i - L_{cap_i})^2 \quad (2)$$

where:

- OI Overload Index
- N_L Number of overloaded lines
- LF_i MVA flow on line i
- L_{cap_i} MVA capacity of line i

2.2 Constraints

2.2.1 Equality Constraints

Generation/load balance Equation

$$\sum_{i=1}^{N_G} P_{gi} - \sum_{i=1}^{N_D} P_{Di} - P_L = 0 \quad (3)$$

2.2.2 Inequality Constraints

(i) Voltage constraints

$$V_{i,\min} \leq V_i \leq V_{i,\max} \quad (4)$$

(ii) Generator constraints

$$P_{gi,\min} \leq P_{gi} \leq P_{gi,\max} \quad (5)$$

$$V_{gi,\min} \leq V_{gi} \leq V_{gi,\max} \quad (6)$$

3 Proposed MODE Algorithm

MODE was proposed by Xue et al. in [18]. This algorithm uses a variant of the original DE, in which the best individual is adopted to create the offspring. A multi-objective-based approach is introduced to implement the selection of the best individuals.

The main algorithm consists of initialization of population, fitness evaluation, Pareto-dominance selection, performing DE operations and reiterating the search on population to reach true Pareto-optimal solutions.

The steps involved in the proposed MODE for transmission line overload management are described below.

Step 1: Set up MODE parameters like population size, number of generations, crossover probability and scaling factor.

Step 2: Read line data, bus data and cost for each generator.

Selection of control variables embedded in the individuals is a first step while applying evolutionary computation algorithm. Generator real powers redispatch and generator bus voltage magnitude is the control variables in this work. Hence, the control variables are generated randomly satisfying their practical operation constraints (5) and (6).

Step 3: For each member of population, run newton raphson (NR) power flow and compute slack bus power and check for limit violations if any. If it violates the operational limit then the corresponding member is regenerated. For each member of population, run NR power flow to evaluate objective functions 1 and 2 using equations (1) and (2). Identify the individuals that give non-dominated solutions in the current population and store them in non-dominated elitist archive (NEA). Set generation counter, $G=0$.

Step 4: Perform mutation and crossover operations using equations (7) and (8) on all the members of the population, i.e., for each parent P_i

$$V_i^{(G)} = X_{r1}^{(G)} + F(X_{r2}^{(G)} - X_{r3}^{(G)}), i = 1, \dots, N_p \quad (7)$$

$$U_i^{(G)} = U_{j,i}^{(G)} = \begin{cases} V_{j,i}^{(G)} & \text{if } \text{rand}_j(0,1) \leq CR \\ x_{j,i}^{(G)} & \text{otherwise} \end{cases} \quad (8)$$

Step 5: Evaluate each member of the population. Check the dominance with its parents. If the candidate dominates the parent, the candidate replaces the parent. If the parent dominates the candidate, the candidate is discarded. Otherwise, the candidate is added to a temporary population (tempPop).

Step 6: Add the latest solution vectors (current population) to the tempPop. Then use the non-dominated sorting and crowding assignment operators to select the individuals to the next generation.

Step 7: Store the non-dominated solutions in the NEA. If NEA size exceeds the desired number of Pareto-optimal set, then select desired number of the least crowded members with the help of crowding assignment operator. Empty the tempPop.

Step 8: Increment the generation counter, G to G+1 and check for termination criteria. If the termination criterion is not satisfied, then go to Step 4; otherwise output the non-dominated solution set from NEA.

Step 9: Apply fuzzy ranking method, determine membership values of the objective functions 1 and 2 using equation (9).

$$\mu(F_i) = \begin{cases} 1; & Fi \leq F_i^{\min} \\ \frac{F_i^{\max} - Fi}{F_i^{\max} - F_i^{\min}}; & F_i^{\min} \leq Fi \leq F_i^{\max} \\ 0; & Fi \geq F_i^{\max} \end{cases} \quad (9)$$

where:

F_i^{\min} and F_i^{\max} are the expected minimum and maximum values of i^{th} objective function.

The value of the membership function indicates how much (in scale from 0 to 1) a solution is satisfying the i^{th} objective F_i . The best solution can then be selected using fuzzy min-max proposition.

Step 10: Determine the best compromise solution of the objective functions 1 and 2 using equation (10).

$$\mu_{bestsolution} = Max\left\{\min[\mu(F_j)]^k\right\} \quad (10)$$

where:

j is number of objectives to be minimized and k are number of Pareto-optimal solutions obtained.

4 Simulation Results

The simulation studies are performed on system having 2.27 GHz Intel 5 processor with 2 GB of RAM in MATLAB environment. The proposed MODE is applied to minimize two conflicting objectives of generation cost and line overload for different contingency cases in IEEE 30 bus system. The transmission line limits and generator cost coefficients are taken from [19]. The upper and lower voltage limits at all the buses except slack are taken as 1.10 p.u and 0.95 p.u respectively. The slack bus voltage is fixed to its specified value of 1.06 p.u. To demonstrate the effectiveness of the proposed method, three different harmful contingency cases are considered which are shown in table 1. The results of all three cases are compared with other evolutionary algorithms. For the studies, the following parameters are used.

Population size: 40

No. of generation: 100

Scaling factor: 0.3

Crossover probability: 0.6.

Table 1. Simulated Cases

| Test system | | Simulated cases |
|-------------|---|--|
| IEEE 30 Bus | A | Outage of line 1-2 under base case |
| | B | Outage of line 1-3 under base case |
| | C | Outage of line 2-5 under 20% increased load case |

The Summary of contingency analysis for the test system before generation rescheduling is summarized in table 2. The control variable setting of the proposed method to minimize the generation cost and line overload for all three cases is shown in table 3. The control variable setting of NSGA-II method to minimize the generation cost and line overload for all three cases is shown in table 4. The control variable setting of the single objective DE method to minimize the line overload for all three cases is shown in table 5. The four intermediate solutions with their membership value out of the obtained non-dominated solution set using the proposed method for all three cases are shown in table 6. The best solutions are shown in bold in table 6 and have a rank of 0.6621, 0.6560 and 0.7813 which means that the two conflicting objectives are satisfied at least 66.21%, 65.60% and 78.13%. The Pareto-optimal solution for the proposed method compared with NSGA-II method for all three cases are shown in table 7.

Table 2. Summary of contingency analysis for IEEE 30 bus system

| Cases | Outage line | Line overloaded | Line limit (MVA) | Actual power flow (MVA) | Overload factor (OLF) | OI | Total power violation (MVA) |
|-------|-------------|-----------------|------------------|-------------------------|-----------------------|-------|-----------------------------|
| A | 1-2 | 1-3 | 130 | 307.0136 | 2.3616 | 61245 | 426.7022 |
| | | 3-4 | 130 | 279.6035 | 2.1508 | | |
| | | 4-6 | 90 | 175.5527 | 1.9506 | | |
| | | 6-8 | 32 | 46.5144 | 1.4536 | | |
| B | 1-3 | 1-2 | 130 | 274.0264 | 2.1079 | 21969 | 196.1237 |
| | | 2-4 | 65 | 86.1203 | 1.3249 | | |
| | | 2-6 | 65 | 92.7203 | 1.4265 | | |
| | | 6-8 | 32 | 35.2567 | 1.1018 | | |
| C | 2-5 | 1-2 | 130 | 213.9041 | 1.6454 | 21190 | 359.8447 |
| | | 1-3 | 130 | 140.0342 | 1.0772 | | |
| | | 2-4 | 65 | 91.2433 | 1.4037 | | |
| | | 3-4 | 130 | 130.0068 | 1.0001 | | |
| | | 2-6 | 65 | 126.3806 | 1.9443 | | |
| | | 4-6 | 90 | 152.5479 | 1.6950 | | |
| | | 5-7 | 70 | 136.8100 | 1.9544 | | |
| | | 6-7 | 130 | 157.6918 | 1.2130 | | |
| 6-8 | 32 | 53.2260 | 1.6633 | | | | |

Table 3. Control variable setting of the proposed method for all three cases

| Control variables | Variable setting | | | | | | | | |
|-------------------|--------------------------------------|--------|--------|--------------------------------------|--------|--------|--|--------|--------|
| | Solution I (Best generation cost) | | | Solution II (Best overload index) | | | Solution III (Best compromise solution) | | |
| | A | B | C | A | B | C | A | B | C |
| P ₁ | 144.90 | 149.42 | 202.02 | 129.67 | 129.63 | 167.59 | 138.16 | 141.38 | 191.36 |
| P ₂ | 57.78 | 54.08 | 52.44 | 64.64 | 60.65 | 42.10 | 63.35 | 57.16 | 47.24 |
| P ₅ | 24.48 | 22.31 | 30.28 | 25.42 | 24.63 | 48.33 | 23.16 | 22.79 | 39.92 |
| P ₈ | 33.24 | 33.02 | 34.44 | 34.57 | 32.20 | 33.89 | 33.77 | 32.69 | 34.18 |
| P ₁₁ | 21.66 | 16.06 | 22.18 | 20.00 | 24.21 | 27.85 | 20.89 | 22.06 | 24.45 |
| P ₁₃ | 16.46 | 19.27 | 20.34 | 21.82 | 21.19 | 35.73 | 18.20 | 17.33 | 21.47 |
| V ₁ | 1.060 | 1.060 | 1.060 | 1.060 | 1.060 | 1.060 | 1.060 | 1.060 | 1.060 |
| V ₂ | 1.044 | 1.035 | 1.054 | 1.020 | 1.034 | 1.028 | 1.026 | 1.034 | 1.047 |
| V ₅ | 1.028 | 0.993 | 0.961 | 0.992 | 0.991 | 0.953 | 1.011 | 1.001 | 0.950 |
| V ₈ | 1.030 | 0.996 | 1.029 | 1.020 | 1.019 | 1.015 | 1.027 | 1.001 | 1.026 |
| V ₁₁ | 1.094 | 1.062 | 1.078 | 1.092 | 1.069 | 1.084 | 1.098 | 1.070 | 1.091 |
| V ₁₃ | 1.076 | 1.040 | 1.078 | 1.054 | 1.056 | 1.045 | 1.045 | 1.041 | 1.100 |

Table 4. Control variable setting of NSGA-II method for all three cases

| Control variables | Variable setting | | | | | | | | |
|-------------------|------------------|--------|--------|-------------|--------|--------|--------------|--------|--------|
| | Solution I | | | Solution II | | | Solution III | | |
| | A | B | C | A | B | C | A | B | C |
| P ₁ | 145.35 | 149.50 | 204.24 | 129.95 | 129.99 | 165.75 | 138.55 | 141.17 | 185.49 |
| P ₂ | 68.01 | 51.23 | 55.02 | 68.26 | 61.60 | 44.95 | 68.22 | 56.78 | 47.36 |
| P ₅ | 24.85 | 23.32 | 29.21 | 27.57 | 26.28 | 49.77 | 25.96 | 24.15 | 39.64 |
| P ₈ | 31.28 | 29.81 | 32.27 | 34.98 | 30.77 | 34.42 | 33.59 | 30.24 | 34.46 |
| P ₁₁ | 14.37 | 23.41 | 23.33 | 14.66 | 26.75 | 21.47 | 14.26 | 24.32 | 21.78 |
| P ₁₃ | 14.61 | 16.73 | 18.08 | 20.61 | 17.04 | 39.94 | 16.77 | 16.78 | 29.45 |
| V ₁ | 1.060 | 1.060 | 1.060 | 1.060 | 1.060 | 1.060 | 1.060 | 1.060 | 1.060 |
| V ₂ | 1.023 | 1.034 | 1.060 | 1.018 | 1.038 | 1.013 | 1.018 | 1.034 | 1.045 |
| V ₅ | 0.997 | 1.023 | 0.969 | 0.993 | 1.025 | 0.952 | 0.994 | 1.022 | 0.985 |
| V ₈ | 1.019 | 1.018 | 1.038 | 1.015 | 1.015 | 0.979 | 1.018 | 1.020 | 1.032 |
| V ₁₁ | 1.079 | 1.070 | 1.095 | 1.071 | 1.090 | 1.079 | 1.085 | 1.070 | 1.100 |
| V ₁₃ | 1.100 | 1.022 | 1.100 | 1.088 | 1.055 | 1.100 | 1.099 | 1.023 | 1.073 |

Table 5. Control variable setting of single objective DE method for all three cases

| Control variables | Variable setting | | | Control variables | Variable setting | | |
|-------------------|------------------|--------|--------|-------------------|------------------|-------|-------|
| | A | B | C | | A | B | C |
| P ₁ | 126.70 | 128.15 | 184.22 | V ₁ | 1.060 | 1.060 | 1.060 |
| P ₂ | 70.89 | 69.61 | 24.81 | V ₂ | 1.024 | 1.033 | 1.035 |
| P ₅ | 26.55 | 25.02 | 48.27 | V ₅ | 0.999 | 0.988 | 0.955 |
| P ₈ | 31.97 | 32.71 | 32.17 | V ₈ | 0.980 | 0.976 | 1.001 |
| P ₁₁ | 16.36 | 10.00 | 29.05 | V ₁₁ | 0.982 | 1.021 | 1.024 |
| P ₁₃ | 24.43 | 28.09 | 37.47 | V ₁₃ | 0.961 | 0.950 | 1.078 |

Table 6. Pareto-optimal intermediate solutions of the proposed method based on fuzzy ranking

| Cases | Generation Cost (\$/h) | OI | μ_1 | μ_2 | μ_{min} |
|-------|------------------------|---------------|---------|---------|---------------|
| A | 843.80 | 1.99 | 0.0837 | 0.9902 | 0.0837 |
| | 841.42 | 66.71 | 0.6621 | 0.6695 | 0.6621 |
| | 843.50 | 4.17 | 0.1561 | 0.9793 | 0.1561 |
| | 841.88 | 40.17 | 0.5500 | 0.8009 | 0.5500 |
| B | 822.65 | 151.42 | 0.7480 | 0.5986 | 0.5986 |
| | 823.66 | 114.17 | 0.6546 | 0.6974 | 0.6546 |
| | 823.21 | 129.79 | 0.6958 | 0.6560 | 0.6560 |
| | 829.39 | 2.18 | 0.1244 | 0.9942 | 0.1244 |
| C | 1058.41 | 305.76 | 0.9592 | 0.3951 | 0.3951 |
| | 1069.89 | 73.19 | 0.7023 | 0.8552 | 0.7023 |
| | 1066.36 | 106.34 | 0.7813 | 0.7896 | 0.7813 |
| | 1072.04 | 62.93 | 0.6543 | 0.8755 | 0.6543 |

Table 7. Pareto-optimal solution for all three cases

| Pareto-optimal solution | Method | Cases | Generation cost (\$/h) | OI |
|-------------------------|----------|-------|------------------------|--------|
| Solution I | Proposed | A | 840.16 | 282.89 |
| | | | NSGA-II | 840.63 |
| | Proposed | B | 819.92 | 377.26 |
| | | | NSGA-II | 821.10 |
| | Proposed | C | 1056.59 | 505.46 |
| NSGA-II | | | 1056.24 | 617.18 |
| Solution II | Proposed | A | 844.15 | 0 |
| | | | NSGA-II | 844.97 |
| | Proposed | B | 830.74 | 0 |
| | | | NSGA-II | 831.44 |
| | Proposed | C | 1101.27 | 0 |
| NSGA-II | | | 1110.51 | 0 |
| Solution III | Proposed | A | 841.42 | 66.71 |
| | | | NSGA-II | 841.73 |
| | Proposed | B | 823.21 | 129.79 |
| | | | NSGA-II | 824.48 |
| | Proposed | C | 1066.36 | 106.34 |
| NSGA-II | | | 1069.09 | 125.68 |

From table 7, it is clear that; overload is managed by changing both rescheduling of generators active power and generator bus voltage magnitude for all three cases. If the operator wants to alleviate the line overload completely, he will choose solution II. However, if the operator allows some overload and takes solution I. To satisfy solutions I and II, the operator will choose solution III which gives best compromise solution. In line 1-2 outage under base load case, GA based approach reported in [20] was not completely minimize the severity index even if rescheduling of generators active power and generator bus voltage magnitude and still has the severity index of 2.473 when compared to proposed method. The control variable setting of GA based

approach to minimize the line overload is shown in table 8. The generation cost and real power loss for best overload index of the proposed method compared with NSGA-II and single objective DE for all three cases are shown in table 9.

Table 8. Control variable setting of GA based approach for case A

| Control variables | Generator active power (MW) | Control variables | Generator bus voltage (p.u) |
|-------------------|-----------------------------|-------------------|-----------------------------|
| P ₁ | 145.49 | V ₁ | 1.035 |
| P ₂ | 57.36 | V ₂ | 0.998 |
| P ₅ | 24.42 | V ₅ | 0.959 |
| P ₈ | 34.82 | V ₈ | 0.967 |
| P ₁₁ | 18.03 | V ₁₁ | 1.02 |
| P ₁₃ | 17.2 | V ₁₃ | 0.9500 |

Table 9. The generation cost and real power loss for best overload index

| Method | Cases | | | | | |
|----------|------------------------|--------|---------|----------------------|-------|-------|
| | Generation Cost (\$/h) | | | Real Power Loss (MW) | | |
| | A | B | C | A | B | C |
| Proposed | 844.15 | 830.74 | 1101.27 | 12.73 | 9.12 | 15.41 |
| NSGA-II | 844.97 | 831.44 | 1110.51 | 12.62 | 9.04 | 16.22 |
| DE | 852.62 | 840.41 | 1112.72 | 13.50 | 10.19 | 15.91 |

The Pareto-optimal front of generation cost and overload index for all three cases compared with NSGA-II method are shown in figure 1, 2 and 3 respectively.

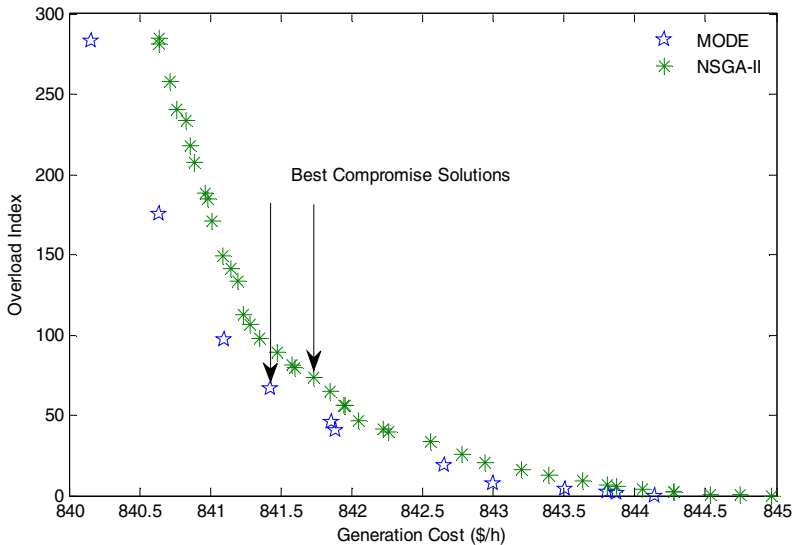


Fig. 1. Pareto-optimal front for Case A

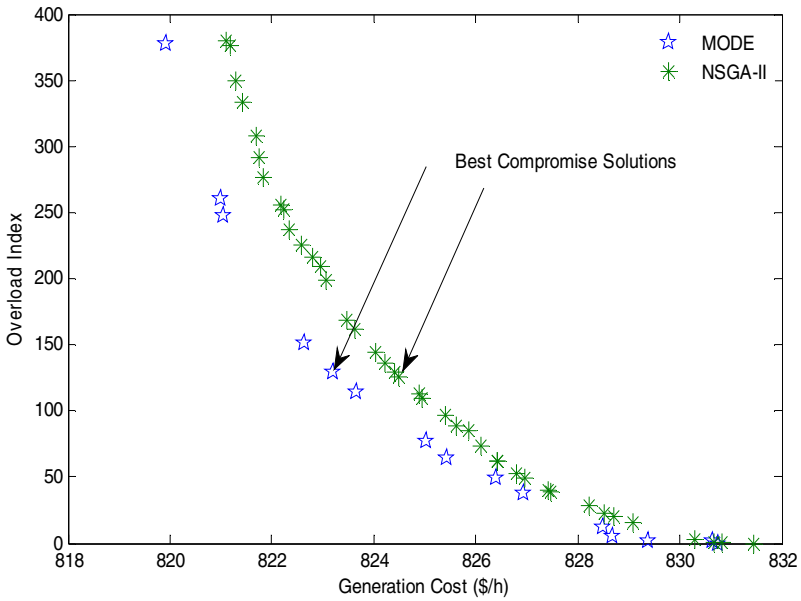


Fig. 2. Pareto-optimal front for Case B

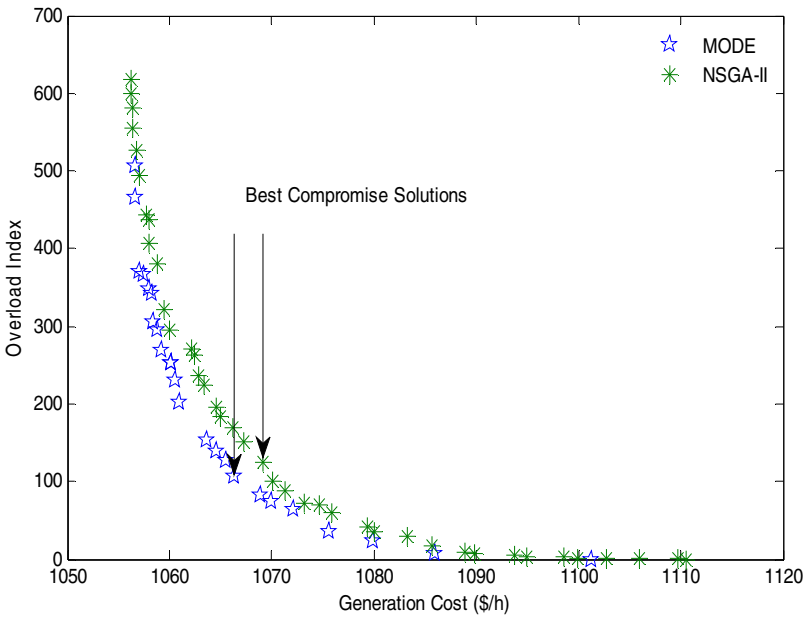


Fig. 3. Pareto-optimal front for Case C

In all three cases, the proposed method relieves all the overloaded lines reported in table 2 by changing both rescheduling of generators active power and generator bus voltage magnitude. From table 9, it is clear that; the proposed method relieves all the overloaded lines with a minimum generation cost when compared to other methods. The computation time for proposed and NSGA-II methods for Case A , Case B and Case C are 22.94, 22.87 and 23.99 and 33.59, 31.30 and 33.98 seconds respectively for 100 generations.

5 Conclusion and Future works

This paper has proposed multi-objective differential evolution based transmission line overload management by both rescheduling of generators active power and generator bus voltage magnitude in a contingent power network. The proposed method has been tested and examined on the standard IEEE-30 bus system. Line overloads are simulated due to unexpected line outage under base case and 20% increased load conditions. In all the considered three cases A, B and C, the proposed method has relieved all the overloaded lines with a minimum generation cost of 844.15 \$/h, 830.74 \$/h and 1101.27 \$/h respectively, when compared to NSGA-II and DE methods. The proposed MODE is capable of handling two conflicting objectives and provides for a set of non-dominated Pareto-optimal solutions with least computation time when compared to NSGA-II. This helps the system operator to select the proper solution for overload alleviation and generation cost minimization whereas, single objective DE algorithm does not provide any choice for the operator and gives only one best solution considering the objectives.

For future works, we aim to extend the proposed approach with Euclidean minimum spanning tree-based multi-objective optimization evolutionary algorithm for overload management in large power system network with inclusion of series FACTS devices along with generation rescheduling and validation using T-test.

References

1. Alsac, O., Scott, B.: Optimal load flow with steady state security. *IEEE Transactions on Power Systems* 93(3), 745–751 (1974)
2. Stott, B., Hobson, E.: Power system security control calculations using linear programming. *IEEE Transactions on Power Systems* 97, 1713–1931 (1978)
3. Todorovski, M., Rajcic, D.: An Initialization Procedure in Solving Optimal Power Flow by Genetic Algorithm. *IEEE Transactions on Power Systems* 21(2), 480–487 (2006)
4. Abido, M.A.: Optimal power flow using particle swarm optimization. *International Journal of Electrical Power and Energy Systems* 24(7), 563–571 (2002)
5. Varadarajan, M., Swarup, K.: Solving multi-objective optimal power flow using differential evolution. *IET, Generation, Transmission and Distribution* 2(5), 720–730 (2008)
6. Duman, S., Guvenc, U., Sonmez, Y., Yoruken, N.: Optimal power flow using gravitational search algorithm. *Energy Conversion and Management* 39, 86–95 (2012)
7. Abido, M.A.: Optimal power flow using tabu search algorithm. *Electric Power Components and Systems* 30 (2002)

8. Sumpavakup, C., Chusanapiputt, S.: A Solution to the Optimal Power Flow Using Artificial Bee Colony Algorithm. In: International Conference on Power System Technology, pp. 1–5 (2010)
9. Udupa, A.N., Purushothama, G.K., Parthasarathy, K., Thukaram, D.: A fuzzy control for network overload alleviation. *International Journal of Electrical Power and Energy Systems* 23(2), 119–128 (2001)
10. Sreejith, S., Psimon, S., Selvan, M.P.: Optimal location of Interline Power Flow Controller in a power system network using ABC algorithm. *Archives of Electrical Engineering* 62(1), 91–110 (2013)
11. Lu, Y., Abur, A.: Static Security Enhancement via Optimal Utilization of Thyristor-Controlled Series Capacitors. *IEEE Transactions on Power Systems* 17(2), 324–329 (2002)
12. Abou, E.L., Ela, A.A., Spea, S.R.: Optimal corrective actions for power systems using multi-objective genetic algorithms. *Electric Power System Research* 79(5), 722–733 (2009)
13. Hazra, J., Sinha, A.K.: Congestion management using multi objective particle swarm optimization. *IEEE Transactions on Power Systems* 22(4), 1726–1734 (2007)
14. Ghahremani, E., Kamwa, I.: Optimal Placement of Multiple-Type FACTS Devices to Maximize Power System Loadability Using a Generic Graphical User Interface. *IEEE Transactions on Power Systems* 28(2), 764–778 (2013)
15. Abido, M.A.: Multi-objective Evolutionary Algorithms for Electric Power Dispatch Problem. *IEEE Transactions on Evolutionary Computation* 11(43) (2006)
16. Gnanambal, K., Babulal, C.K.: Maximum loadability limit of power system using hybrid differential evolution with particle swarm optimization. *Electrical Power and Energy Systems* 43(1), 150–155 (2012)
17. Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P.N., Zhangd, Q.: Multi-objective Evolutionary Algorithms: A Survey of the State-of-the-art. *Swarm and Evolutionary Computation* 1(1), 32–49 (2011)
18. Xue, F., Sanderson, A.C., Graves, R.J.: Multi-objective-based multi-objective differential evolution. In: *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, Canberra, Australia, vol. 2, pp. 862–869 (2003)
19. Zimmerman, R.D., Murillo-Sanchez, C.E., Gan, D.(D.): MATPOWER: A MATLAB Power System Simulation Package, <http://www.pserc.cornell.edu/matpower/>
20. Narmatha Banu, R., Devaraj, D.: Multi-objective Evolutionary Algorithm for Security Enhancement. *Journal of Electrical Systems* 5(4) (2009)

Normalized Normal Constraint Algorithm Based Multi-objective Optimal Tuning of Decentralised PI Controller of Nonlinear Multivariable Process – Coal Gasifier

Rangasamy Kotteswaran¹ and Lingappan Sivakumar²

¹ Department of Instrumentation and Control Engineering,
St. Joseph's College of Engineering, Chennai, India
kotteswaran@gmail.com

² Formerly General Manager(Corporate R&D), BHEL, Hyderabad, Presently with
Sri Krishna college of Engineering and Technology, Coimbatore, Tamil Nadu, India
lingappansivakumar@gmail.com

Abstract. Almost all the industrial processes are multivariable in nature and are very difficult to control, since it involves many variables, strong interactions and nonlinearities. Conventional controllers are most widely used with its optimal parameters for such processes because of its simplicity, reliability and stability. Coal gasifier is a highly nonlinear multivariable process with strong interactions among the loop and it is difficult to control at 0% operating point with sinusoidal pressure disturbance. The present work uses Normalized Normal Constraint (NNC) algorithm to tune the parameters of decentralised PI controller of coal gasifier. Maximum absolute error (AE) and Integral of Absolute Error (IAE) are objective function while the controller parameters of decentralised PI controller are the decision variables for the NNC algorithm. With the optimal controller the coal gasifier provides better response at 0%, 50% and 100% operating points and also the performance tests shows good results.

Keywords: Coal gasifier, Multi-Objective Optimization, Multivariable process, Normalized Normal Constraint Algorithm, PID Controller tuning.

1 Introduction

Gasification is a thermo-chemical process, that convert any carbonaceous material (Solid Fuel-coal) in to combustible gas known as "producer gas or syngas" under certain pressure and temperature. Coal gasifier is a highly non-linear, multivariable process, having five controllable inputs, few non-control inputs and four outputs with a high degree of cross coupling between them. The process is a four-input, four output regulatory problem for the control design (keeping limestone at constant value). Gasifier exhibits very complex dynamic behaviour with mixed fast and slow dynamics and it is highly difficult to control. The plant inputs and outputs with their allowable limits, control specifications are mentioned in the challenge pack [1]. The performance of coal gasifier under sinusoidal pressure disturbance at 0% operating

point does not satisfy the performance requirement as mentioned in [1]. Until recently researchers have attempted to design controllers and/or retuned the baseline controller to meet the performance requirements at 0%, 50% and 100% load conditions. Analysis, design and implementations of advanced control schemes for coal gasifier are reported in the literature [2-9]. Apart from conventional techniques soft computing techniques are also utilized [10 - 12], where controller parameters are returned to meet the desired objectives. This provides the scope for new optimization algorithms to be used with this problem.

2 Control Specifications

The complete transfer function model of the gasifier can be represented in the form:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} & G_{15} \\ G_{21} & G_{22} & G_{23} & G_{24} & G_{25} \\ G_{31} & G_{32} & G_{34} & G_{34} & G_{35} \\ G_{41} & G_{42} & G_{43} & G_{44} & G_{45} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} + \begin{bmatrix} G_{d1} \\ G_{d2} \\ G_{d3} \\ G_{d4} \end{bmatrix} \times d \tag{1}$$

Where,

G_{ij} =transfer function from i^{th} input to j^{th} output

- y_1 = fuel gas caloric value (J/kg);
- y_2 =bed mass (kg);
- y_3 =fuel gas pressure (N/m²);
- y_4 =fuel gas temperature (K);
- u_1 =char extraction flow (kg/s);
- u_2 =air mass flow (kg/s);
- u_3 =coal flow (kg/s);
- u_4 =steam mass flow (kg/s);
- u_5 =limestone mass flow (kg/s);
- d =sink pressure (N/m²);

Limestone flow rate is fixed at 1/10th of coal flow rate and thus the process can be reduced to 4X4 MIMO process for control purpose. For a multivariable process decentralised control schemes are usually preferred. The structure of decentralized controller used in gasifier control can be represented as;

$$G_c(s) = \begin{pmatrix} 0 & \left(K_p + \frac{1}{\tau_i s} \right) & 0 & 0 \\ K_f & 0 & K_p & 0 \\ 0 & 0 & 0 & \left(K_p + \frac{1}{\tau_i s} \right) \\ \left(K_p + \frac{1}{\tau_i s} \right) & 0 & 0 & 0 \end{pmatrix} \tag{2}$$

Where, K_p = proportional gain;

τ_i = Integral time;

K_f = feedforward gain.

It employs three PI controllers and one feedforward + feedback controller for coal flow rate. The given controller structure with provided controller parameters satisfies the performance requirements at 50% and 100% operating points but fails to satisfy the constraints at 0% load for sinusoidal pressure disturbance(i.e. PGAS exceeds the limit of ±0.1bar). The decentralised controller may be re-tuned to meet the desired performance requirement even at 0% operating point.

2.1 Input Limits

The input actuator flow limits and rate of change of limit are associated with the physical properties of the actuator, should not exceed as shown in table 1.

Table 1. Input limits

| Input variable | Max(kg s ⁻¹) | Min(kg s ⁻¹) | Rate(kg s ⁻²) |
|-------------------------|--------------------------|--------------------------|---------------------------|
| Coal inlet flow (WCOL) | 10 | 0 | 0.2 |
| Air inlet flow (WAIR) | 20 | 0 | 1.0 |
| Steam inlet flow (WSTM) | 6.0 | 0 | 1.0 |
| Char extraction (WCHR) | 3.5 | 0 | 0.2 |

2.2 Output Limits

Gasifier outputs should be regulated within the limits (table 2) for sink pressure (PSink) disturbance test, load change test and other tests. The desired objective is the outputs should be regulated as closely as possible to the demand.

Table 2. Output limits

| Output variable | Objective | Limits |
|---------------------------------|------------------|-------------------------|
| Fuel Gas Calorific vale (CVGAS) | Minimize | ± 10KJ kg ⁻¹ |
| Bed mass (MASS) | fluctuations | ± 500 kg |
| Fuel Gas Pressure (PGAS) | For all | ± 0.1 bar |
| Fuel Gas Temperature (TGAS) | Output variables | ± 1 K |

3 Multiobjective Optimization

In the recent past Multi objective optimization algorithm [13,14] are most widely used in process industries than single objective optimization algorithm since the design requirements are more. Multi-Objective optimization involves two or more objectives are optimized simultaneously under certain constraints. The discussions about various multi-objective evolutionary approaches from the analytical weighted aggression to population based approaches, and the Pareto-optimality concepts are discussed in literature. Pareto based approaches are most suitable for multi-objective optimization problems, due to the ability to produce multiple solutions in less computation time. Non-Dominated Sorting Genetic Algorithm-II (NSGA-II), Pareto Archive Evolutionary Strategy (PAES) and micro Genetic algorithm (microGA) are the three highly competitive Evolutionary Multi Objective (EMO) algorithms used in recent past. The mathematical formulation of Normalized Normal Constraint (NNC) algorithm was described in [15]. Some of the characteristics of this algorithm may include; Initial condition assumed to the optimization routine dominates the NNC results. This algorithm does not have memory of the pareto points obtained

previously. The minimization of desired performance specification in multi objective problems can be states as

$$\min[\mu_1(x) \mu_2(x) \dots \dots \dots, \mu_n(x)] \tag{3}$$

Subject to

$$\begin{aligned} g_q(x) &\leq 0, & (1 \leq q \leq r) \\ h_k(x) &= 0, & (1 \leq k \leq s) \\ x_{li} &\leq x_i \leq x_{ui}, & (1 \leq i \leq n_x) \end{aligned} \tag{4}$$

Where, $g_q(x) = r$ inequality constraint; $h_k(x) = s$ equality constraint
 $x = n_x$ dimension vector of design variables; x_{li} = lower constraint limit
 x_{ui} = upper constraint limit

In our proposed work, the objective function is to minimize the 8 variables as listed in table 3. The controller parameters of decentralized PI controller for all the four loops are the decision variables (equation 2). A unique solution is not possible with this formulation. Minimize the each objective function,

$$\min_x \mu_i(x) \quad i = 1, 2, \dots, n \tag{5}$$

The ends of pareto frontier is the function of obtained anchor point (figure 1).

The Utopian point is given by μ^u ,

$$\mu^u = [\mu_1(x^{1*}) \mu_2(x^{2*}) \dots \dots \dots, \mu_n(x^{n*})]^T \tag{6}$$

Normalization of searching space is given by

$$L = \{l_1 \ l_2 \ \dots \dots \ l_n\} = \mu^N - \mu^u \tag{7}$$

Where,
$$\mu^N = [\mu_1^N \ \mu_2^N \ \dots \dots \dots, \mu_n^N] \tag{8}$$

$$\mu_i^N = \max[\mu_i(x^{1*}) \ \mu_i(x^{2*}) \ \dots \dots \dots, \mu_i(x^{n*})] \tag{9}$$

And thus the normalized design metrics is given by

$$\bar{\mu}_i = \frac{\mu_i - \mu_i(x^{1*})}{l_i}, \quad i = (1, 2, \dots, n) \tag{10}$$

The difference between normalized anchor vectors is given by

$$\bar{N}_k = \mu^{-n*} - \mu^{-k*} \tag{11}$$

For a prescribed number of solutions m_k , the normalized increment δ_k is defied along the direction \bar{N}_k

$$\delta_k = \frac{1}{1 - m_k}, \quad (1 \leq k \leq n - 1) \tag{12}$$

The distributed points on the Utopian hyperplane are described as

$$\bar{X}_{pj} = \sum_{k=1}^n \alpha_{kj} \mu^{-k*} \tag{13}$$

Where,
$$\sum_{k=1}^n \alpha_{kj} = 1 \quad \text{and} \quad 0 \leq \alpha_{kj} \leq 1 \tag{14}$$

And thus the multi-objective optimization problem can be transformed into minimization of X_{pj} single objective problem in normalization domain.

i.e.
$$\min_x \bar{\mu}_n \tag{15}$$

subject to

$$\begin{aligned}
 g_q(x) &\leq 0, & (1 \leq q \leq r) \\
 h_k(x) &= 0, & (1 \leq k \leq s) \\
 x_{li} \leq x_i \leq x_{ui}, & & (1 \leq i \leq n_x)
 \end{aligned}
 \tag{16}$$

$$\begin{aligned}
 \bar{N}_k^T (\bar{\mu} - \bar{X}_{pj}) &\leq 0 \\
 \bar{\mu} &= [\bar{\mu}_1(x), \dots, \bar{\mu}_n(x)]^T
 \end{aligned}
 \tag{17}$$

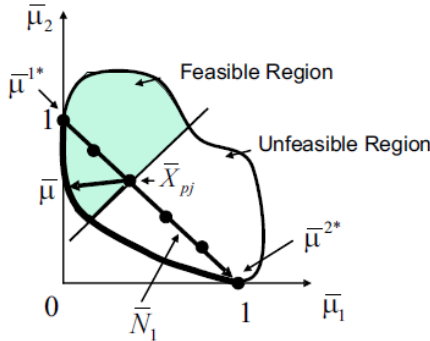


Fig. 1. Graphical representation for Normal constraint method

4 Problem Formulation and Implementation

Figure 2 shows the implementation of Multi-objective Optimization technique applied to tune the parameters of PI controller of ALSTOM gasifier. Integral of Absolute Error (IAE) and Maximum Absolute Error (AE) for each output at 0% load and 0% change in coal quality are the objective function for Multiobjective NNC algorithm while controller parameters of PI controller are taken as decision variables.

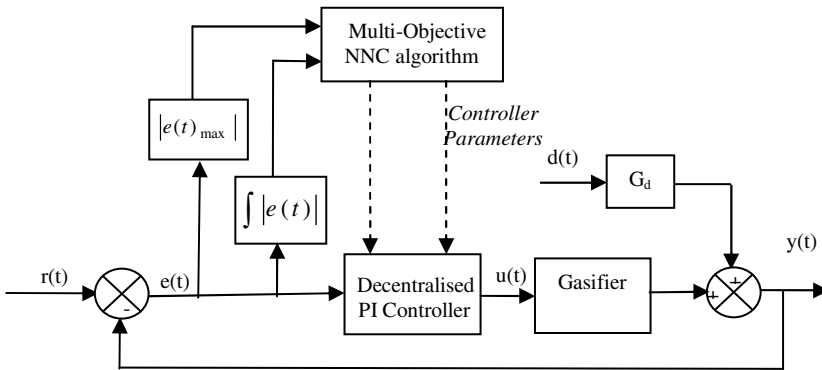


Fig. 2. Block diagram of Optimization scheme

Table 3 shows the desired 8 objective functions to be used for solving the multi-objective problem. Input constraints are associated with the given Simulink model and it is not included in the specifications. Minimization of these 8 desired performance specifications is the objective function ($\mu_n(x)$). The parameters of multivariable baseline PI controller for the four loops are the decision variables ($x=8$). The upper (x_{ui}) and lower (x_{li}) boundary limits are set as 1 and 0 respectively. The controller should respond quickly than the process and hence sampling time is selected as 0.5 seconds.

Table 3. Objectives

| Objectives | Description |
|------------|---|
| 1 | Max. absolute error of CVGAS ($\mu_1(x)$) |
| 2 | Max. absolute error of MASS($\mu_2(x)$) |
| 3 | Max. absolute error of PGAS($\mu_3(x)$) |
| 4 | Max. absolute error of TGAS($\mu_4(x)$) |
| 5 | IAE of CVGAS over 300s($\mu_5(x)$) |
| 6 | IAE of MASS over 300s($\mu_6(x)$) |
| 7 | IAE of PGAS over 300s($\mu_7(x)$) |
| 8 | IAE of TGAS over 300s($\mu_8(x)$) |

The procedure is as follows;

- 1) At 0% load and 0% coal quality apply a sinusoidal pressure disturbance (amplitude 0.2bar and frequency of 0.04Hz).
- 2) Run the simulation over 300seconds.
- 3) Calculate IAE and AE (objective function $\mu_n(x)$ as shown in table 3).
- 4) Run NNC algorithm (Matlab code). Upper and lower constraint limits are fixed as 1 and 0 respectively.
- 5) Best optimal controller parameters are obtained. These controller parameters (decision variables) of PI controller are the best tuned values.

The decentralized controller parameters (K_p , K_f and K_i) of the four loops (CV- Calorific value, BM-bedmass, Pr – pressure and Tg – temperature) by different approaches are given in table 4. These parameters are used to evaluate the performance of performance of the gasifier under different scenarios.

Table 4. Comparison of PI Controller parameters

| Parameter | Dixon PI[1] | Simm A[10] | MOPI[12] | SOPI[12] | NNC-PI |
|-----------|-------------|-------------|------------|-----------|---------|
| CV_Kp | -0.000123 | -0.00015445 | -0.016972 | -0.01956 | -0.0003 |
| CV_Ki | -0.0000804 | -0.00010867 | -0.024813 | -0.05001 | -0.0008 |
| BM_Kp | 0.1451 | 0.1814 | 0.18498 | 0.119 | 0.26063 |
| BM_Kf | 1.0328 | 1.2910 | 1.741 | 1.029 | 1.82638 |
| Pr_Kp | 0.000202 | 0.00022281 | 0.0003055 | 0.0002575 | 0.0002 |
| Pr_Ki | 0.0000265 | 0.00000614 | 0.00001077 | 0 | 0.00001 |
| Tg_Kp | 1.7013 | 2.1266 | 2.2825 | 2.0420 | 1.69774 |
| Tg_Ki | 0.00948 | 0.0119 | 0.097237 | 0.2220 | 0.01 |

5 Performance Tests

Following performance tests are conducted to verify the robustness of the system for the tuned values of baseline PI controller. Test results should satisfy the constraints for all performance tests. Using the tuned parameters, simulation is run for 300 seconds at 0%, 50% and 100% load conditions with sinusoidal and step pressure disturbance and any constraint violations are observed.

5.1 Pressure Disturbance Tests

A step change in pressure disturbance of 0.2 bar and a sinusoidal pressure disturbance of amplitude 0.2 bar and frequency 0.04 Hz is applied to the Alstom gasifier at 0%, 50% and 100% load conditions.

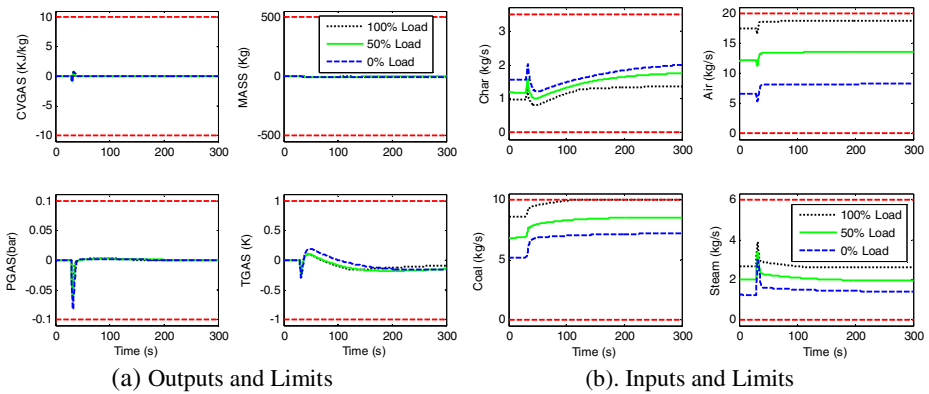


Fig. 3. Response to step disturbance at 0%, 50% and 100% load

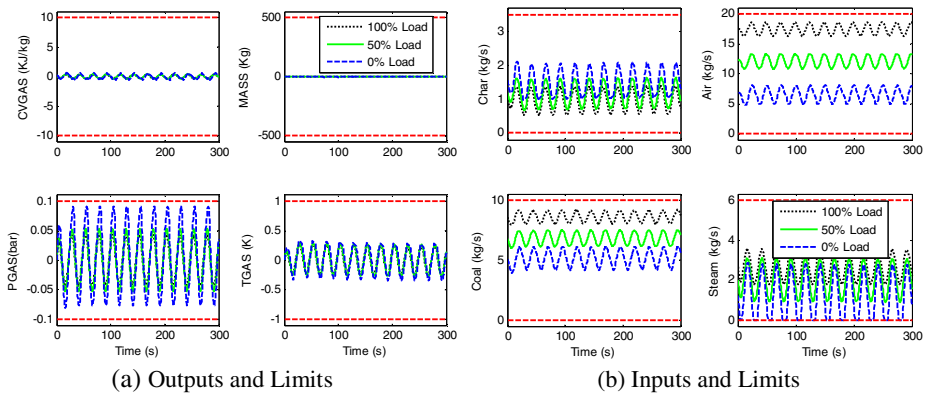


Fig. 4. Response to sinusoidal disturbance at 0%, 50% and 100% load

The obtained results for the above six pressure disturbance tests are compared with [1] and listed in table 5.

Table 5. Summary of pressure disturbance test

| Test Description | Output | Maximum Absolute Error | | IAE | |
|-----------------------------------|-------------------------|------------------------|----------|------------|------------|
| | | NNC-PI | Dixon-PI | NNC-PI | Dixon-PI |
| 100% Load, Step Disturbance | CVGAS(J/kg) | 925.28 | 4885.23 | 7391.13 | 60989.48 |
| | MASS(kg) | 6.94 | 6.94 | 1622.00 | 1597.03 |
| | PGAS(N/m ²) | 5713.84 | 5018.94 | 89738.61 | 78475.47 |
| | TGAS(°K) | 0.24 | 0.24 | 58.09 | 65.09 |
| 50% Load, Step Disturbance | CVGAS(J/kg) | 965.75 | 5102.16 | 8071.78 | 64766.48 |
| | MASS(kg) | 8.45 | 8.45 | 3041.31 | 840.04 |
| | PGAS(N/m ²) | 6433.74 | 5790.93 | 112756.06 | 94310.73 |
| | TGAS(°K) | 0.25 | 0.27 | 73.23 | 77.13 |
| 0% Load, Step Disturbance | CVGAS(J/kg) | 946.34 | 5875.95 | 10128.00 | 86561.16 |
| | MASS(kg) | 11.05 | 11.05 | 3183.29 | 1330.92 |
| | PGAS(N/m ²) | 8326.24 | 7714.53 | 117495.57 | 120167.73 |
| | TGAS(°K) | 0.29 | 0.32 | 65.51 | 77.05 |
| 100% Load, Sinusoidal Disturbance | CVGAS(J/kg) | 353.39 | 4101.30 | 132962.88 | 1545471.04 |
| | MASS(kg) | 9.39 | 10.89 | 4066.29 | 4154.65 |
| | PGAS(N/m ²) | 4366.07 | 4981.41 | 1616664.37 | 1857629.38 |
| | TGAS(°K) | 0.27 | 0.38 | 90.75 | 134.44 |
| 50% Load, Sinusoidal Disturbance | CVGAS(J/kg) | 397.32 | 4715.68 | 149245.74 | 1759740.23 |
| | MASS(kg) | 11.04 | 12.87 | 4930.16 | 5041.36 |
| | PGAS(N/m ²) | 5336.92 | 6209.91 | 1983626.23 | 2307614.42 |
| | TGAS(°K) | 0.30 | 0.42 | 100.28 | 149.47 |
| 0% Load, Sinusoidal Disturbance | CVGAS(J/kg) | 666.55 | 5869.69 | 185010.85 | 2074977.65 |
| | MASS(kg) | 14.26 | 16.35 | 6240.53 | 6016.65 |
| | PGAS(N/m ²) | 9116.02 | 11960.42 | 3071406.23 | 3845931.81 |
| | TGAS(°K) | 0.34 | 0.48 | 119.79 | 159.09 |

5.2 Load Change Test

Stability of the gasifier and controller function across the working range of the plant is verified by load change test. For this purpose the system is started at 50% load in steady state and ramped it to 100% over a period of 600 seconds (5% per minute). The actual load, CVGAS and PGAS track their demands quickly to setpoint while Bedmass takes more time to reach its steady state, though manipulated inputs coal flow and char flow have reached their steady state immediately. TGAS reached its steady state at around 12 minutes from the start, immediately char flow rate has regulated back nearly to its steady state point,

5.3 Coal Quality Test

The quality of coal gas depends on the coal quality (carbon content and moisture content). In this test, the quality of coal increased and decreased by 18%, and the above pressure disturbance test are conducted to verify the robustness of the controller. Input-output responses for sinusoidal and step change in PSink are obtained for 300 seconds and are shown in figure 5 to 10.

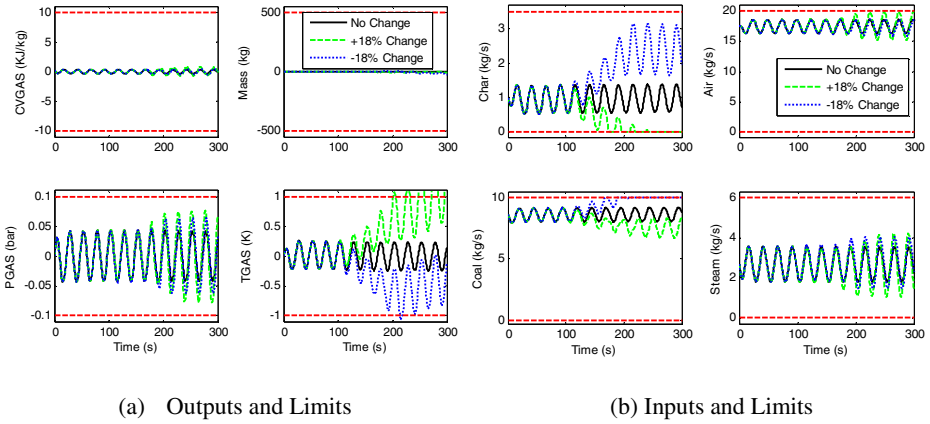


Fig. 5. Response to change in Coal quality at 100 % Load for sinusoidal change in PSink

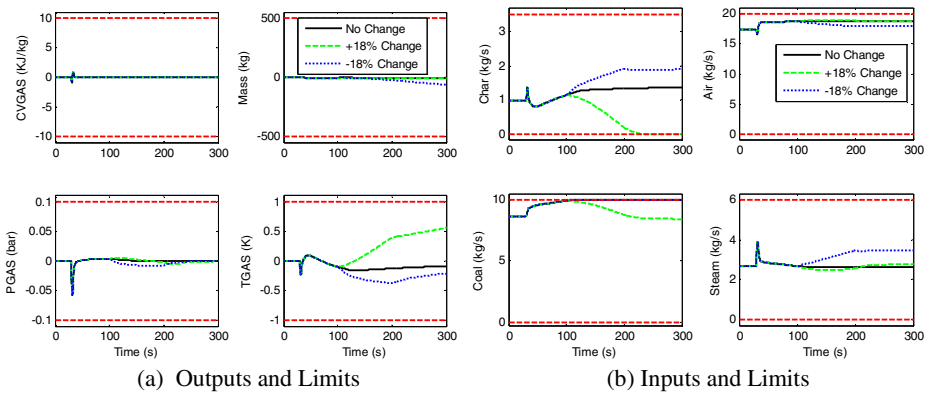


Fig. 6. Response to change in Coal quality at 100 % Load for step change in PSink

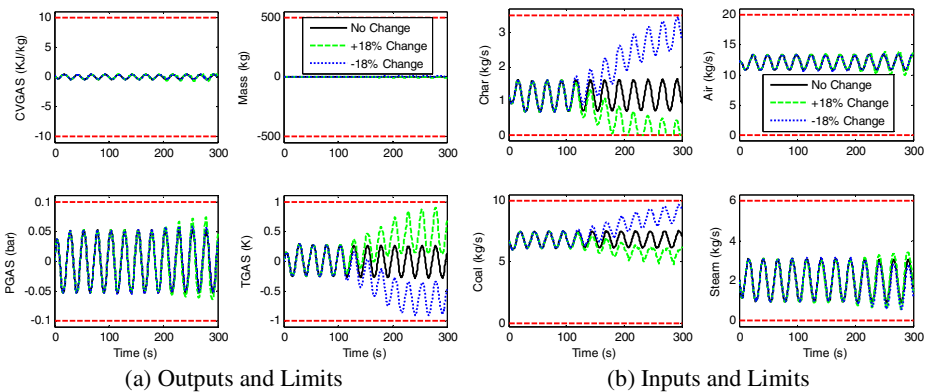


Fig. 7. Response to change in Coal quality at 50% Load for sinusoidal change in PSink

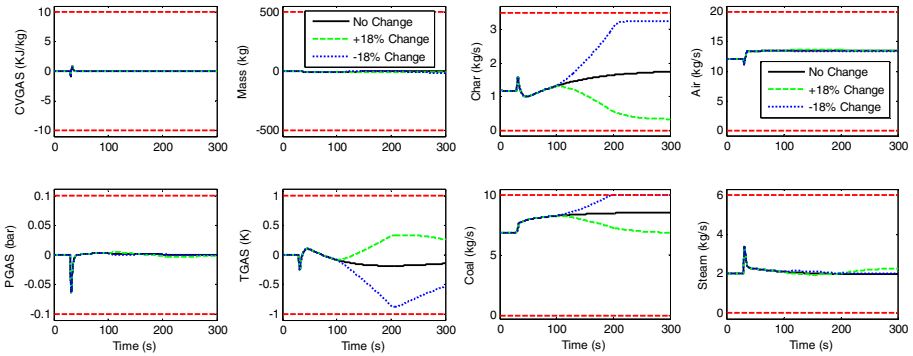


Fig. 8. Response to change in Coal quality at 50 % Load for step change in PSink

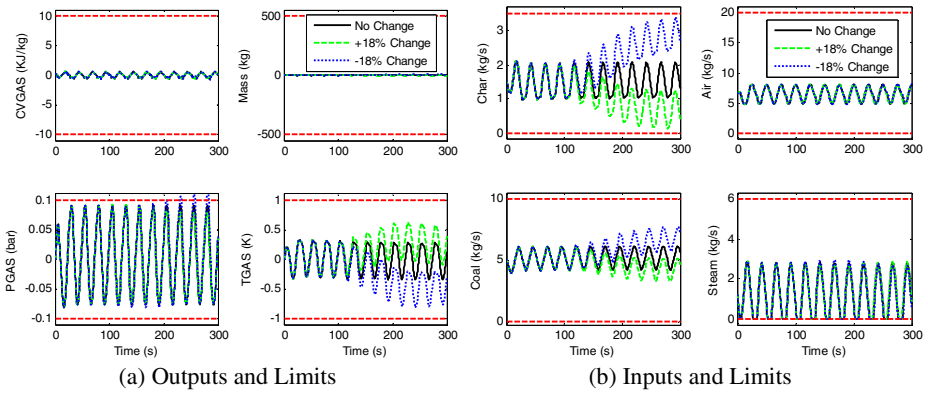


Fig. 9. Response to change in Coal quality at 0 % Load for sinusoidal change in PSink

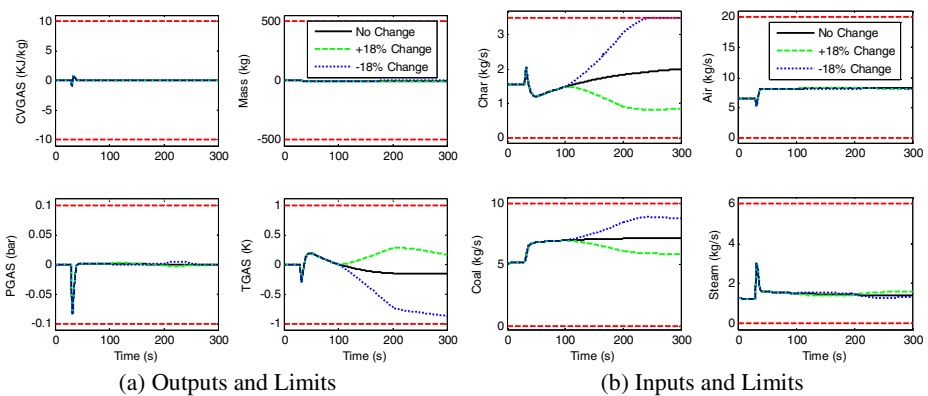


Fig. 10. Response to change in Coal quality at 0% Load for step change in PSink

The analysis of the above test is shown in table 6, which shows the violation of the variables under positive and negative change in coal quality. Since input constraints are inbuilt in the actuator limits, output constraints are considered to be the actual violation. TGAS and PGAS violate the limits under change in coal in coal quality for sinusoidal pressure disturbance and no output variable is found for step pressure disturbance.

Table 6. Violation variables under coal quality change ($\pm 18\%$) (\uparrow - the variable reaches its upper limit, \downarrow the variable reaches its lower limit)

| Load | 100% | | 50% | | 0% | |
|------------------------------|--------------------------------------|-------------------|--------------------------------------|-----------------|--|-----------------|
| Disturbance type | Sine | Step | Sine | Step | Sine | Step |
| Coal quality increase (+18%) | Char \downarrow Tgas \uparrow | Char \downarrow | Char \downarrow Tgas \uparrow | Within limits | Char \downarrow WStm \downarrow Pgas \uparrow | Within limits |
| Coal quality decrease (-18%) | Coal \uparrow Tgas \downarrow | Coal \uparrow | Within limits | Coal \uparrow | Char \uparrow Pgas \uparrow WStm \downarrow | Char \uparrow |

6 Conclusion

This paper uses Normalized Normal Constraint (NNC) algorithm to retune the parameters of decentralised PI controller for pressure loop of coal gasifier. The existing controller with decentralised PI controller does not satisfy the performance requirements at 0% operating point for sinusoidal disturbance and hence optimal tuning parameters are required. Proportional gain and Integral time for the decentralised PI controller are the decision variables while the maximum Absolute Error (AE) and Integral of Square Error (IAE) are the objective function for NNC algorithm. The existing PI controller parameters are replaced by obtained controller parameters and performance tests are conducted. Pressure disturbance test shows excellent results and meets the performance requirement satisfactorily even at 0% operating point. Load change test and coal quality tests are also conducted. The limits for coal quality variation are set to $\pm 18\%$. For the allowable limits of coal quality variations test results shows that the NNC based decentralised PI controller provides good results in all aspects.

Acknowledgement. The authors would like to thank Dr.Roger Dixon, Director of Systems Engineering Doctorate Centre, Head of Control Systems Group, Loughborough University, UK for useful communication through email, and the managements of St. Joseph’s College of Engineering, Chennai and Sri Krishna College of Engineering & Technology, Coimbatore for their support.

References

[1] Dixon, R., Pike, A.W.: Alstom Benchmark Challenge II on Gasifier Control. IEE Proceedings - Control Theory and Applications 153(3), 254–261 (2006)
 [2] Chin, C.S., Munro, N.: Control of the ALSTOM gasifier benchmark problem using H2 methodology. Journal of Process Control. 13(8), 759–768 (2003)

- [3] Al Seyab, R.K., Cao, Y., Yang, S.H.: Predictive control for the ALSTOM gasifier problem. *IEE Proceedings - Control Theory and Application* 153(3), 293–301 (2006)
- [4] Al Seyab, R.K., Cao, Y.: Nonlinear model predictive control for the ALSTOM gasifier. *Journal of Process Control* 16(8), 795–808 (2006)
- [5] Agustriyanto, R., Zhang, J.: Control structure selection for the ALSTOM gasifier benchmark process using GRDG analysis. *International Journal of Modelling, Identification and Control* 6(2), 126–135 (2009)
- [6] Tan, W., Lou, G., Liang, L.: Partially decentralized control for ALSTOM gasifier. *ISA Transactions* 50(3), 397–408 (2011)
- [7] Sivakumar, L., Anitha Mary, X.: A Reduced Order Transfer Function Models for Alstom Gasifier using Genetic Algorithm. *Int. J. of Computer Applications* 46(5), 31–38 (2012)
- [8] Kotteeswaran, R., Sivakumar, L.: Lower Order Transfer Function Identification of Nonlinear MIMO System-Alstom Gasifier. *International Journal of Engineering Research and Applications* 2(4), 1220–1226 (2012)
- [9] Huang, C., Li, D., Xue, Y.: Active disturbance rejection control for the ALSTOM gasifier benchmark problem. *Control Engineering Practice* 21(4), 556–564 (2013)
- [10] Simm, A., Liu, G.P.: Improving the performance of the ALSTOM baseline controller using multiobjective optimization. *IEE-Control Theory and Applications* 153(3), 286–292 (2006)
- [11] Nobakhti, A., Wang, H.: A simple self-adaptive Differential Evolution algorithm with application on the ALSTOM gasifier. *Applied Soft Computing* 8(1), 350–370 (2008)
- [12] Xue, Y., Li, D., Gao, F.: Multi-objective optimization and selection for the PI control of ALSTOM gasifier problem. *Control Engineering Practice* 18(1), 67–76 (2010)
- [13] 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)
- [14] Zhao, S.Z., Willjuice Iruthayarajan, M., Baskar, S., Suganathan, P.N.: Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization. *Information Sciences* 181(16), 3323–3335 (2011)
- [15] Messac, A., Ismail-Yahaya, A., Mattson, C.A.: The Normalized Normal Constraint Method for Generating the Pareto Frontier. *Structural and Multidisciplinary Optimization* 25(2), 86–98 (2003)

Simulated Annealing Based Real Power Loss Minimization Aspect for a Large Power Network

Syamasree Biswas (Raha), Kamal Krishna Manadal, and Niladri Chakraborty

Dept. of Power Engineering, Jadavpur University, Saltlake 2nd Campus, Kolkata-98
syamasree@gmail.com

Abstract. Real field power systems are suffering from the various problems since two decades passed. Among them one of the vital problems is the real power loss minimization issue. In this paper the said issue is tried to be solved utilizing one of the interesting meta-heuristic technique i.e., Simulated Annealing method. While solving the same, few control and state variables are controlled and monitored such that system parametric violations do not occur. Finally obtained results are compared with other reported technique which proves the effectiveness of the approached technique.

Keywords: Real power loss minimization, Reactive power dispatch, Voltage Stability, Simulated Annealing.

1 Introduction

Real Power Loss Minimization (RPLM) is an all time important aspect of the power system. It is handled strictly as real power loss reduces the efficiency of power transmission & distribution. Initially this problem is solved as optimal power flow [2] which evolved many factors like cost minimization, real power loss minimization etc. Cost minimization is always prioritized over loss minimization. Hence the real power loss minimization is handled separately as reactive power dispatch (RPD) problem. The RPD problem focuses the real power loss minimization aspect along with the stable voltage profile [3]. The RPD problem is evaluated by controlling generator bus voltages and variable transformer tap changer along with the shunt capacitors placements to few load buses. Alike the various optimization problem this problem are also having few control and state variables. The control variables are the Generator bus voltages (V_G), Shunt Var compensators (Q_C), and Transformer taps setting (T). The state variables for the considered problem are Generator real power output at slack bus (P_{g1}), Generator reactive power output (Q_G), Load bus voltage (V_L), and bus voltage angle (θ). The RPD problem has been solved by several linear and non-linear programming since two decades are passed. As the nature of the RPD problem is nonlinear, non differentiable and multi-constraint based, perfection in obtaining the optimum result remains untouched while solving with the previously mentioned techniques. In this connection meta-heuristics techniques handle this type of problem quite accurately [12]. These techniques have already achieved successful feedback in solving the RPD problem while testing with the bus systems like IEEE 6-bus, IEEE 14-bus and IEEE 30-bus

system. Among them few meta-heuristics techniques are tabu search algorithm [1], evolutionary programming [15], genetic algorithm [9], modified genetic algorithm [11], differential evolutionary algorithm [5, 13], particle swarm optimization [7] and simulated annealing [18] which have been observed to be very effective. In this paper single solution based Simulated Annealing (SA) technique is studied to solve the said problem for the higher bus systems like IEEE 118-bus system. In 1983 Kirpatrik et al. first proposed the said technique which is based on the annealing phenomenon of a solid, liquid or gaseous material [6]. The benefit of the technique is not only its fastness it can escape from the local minima. Population based techniques fit well for the RPD problem although it requires large processing time. Therefore sometimes SA is utilised with many other population based methods like GA as a hybrid solution methodology (GASA) [17]. In this paper SA technique gives much better result compared to the base cases where Newton Raphson method is applied to solve the RPD problem without shunt capacitors placement. Additionally SA based result are compared with few other population based evolutionary techniques which shows improvement in optimum value as well as operating time.

2 Problem Formulation:

RPD problem explore the minimization of the real power loss by maintaining stable voltage profile. In this paper RPD is solved by controlling V_G , Q_C and T . Apart from the controlled variable few state variables are to be measured. Controlled variables are generated with the help of evolutionary algorithms whereas state variable are bounded by the system operating limit. These are maintained by few equality and inequality equations during the programme execution.

In the generalized approach, the RPD problem can be expressed as:

$$\min f(x,u) \text{ s.t.}$$

$$\left. \begin{aligned} g(x,u) &= 0 \\ h(x,u) &\leq 0 \end{aligned} \right\} \quad (1)$$

Where $f(x,u)$ is the objective function which explains the minimization of total active power losses in transmission system (RPD) where x represents state or dependent variable of the system $[\theta, V_L, P_{gI}, Q_G]^T$ and u represents the control or independent variable $[V_G, T, Q_C]^T$.

In this connection, the tuned RPD problem can be expressed as:

$$\min P_{Loss} = \sum_{i=1}^{ng} P_{GEN} - \sum_{i=1, i \neq \text{slack bus}}^{nbus} P_{LOAD} \quad (2)$$

Where, ng is the total number of generator buses and $nbus$ are the total buses in the network. P_{GEN} and P_{LOAD} is the total generated power and the total load power connected to the given system. P_{Loss} is the power loss in the network. In this paper P_{GEN} and P_{LOAD} is calculated via Newton Raphson method.

The minimization of the above optimisation problem is subjected to a number of equality and inequality constraints. Equality constraints are comprised of active and reactive power balance equation. Inequality constraints are consisted of Generation constraints which further are represented as Generation bus voltages, reactive power outputs, Transformer constraints, Shunt VAR constraints and Security Constraints. Due to large size of the system, few simplifications in the objective function are considered without violating the aim. Once the objective function is settled simulated annealing (SA) technique is applied to solve the RPD problem.

3 Simulated Annealing (SA)

In 1983 being inspired by the annealing process of liquids to freeze or metals and glass to crystallize, Kirkpatrick, Gelatt and Vecchi proposed a new solution methodology named Simulated Annealing Technique [6]. SA has several strong points over other metaheuristics techniques. Amongst them the most important one is that the SA will never stick into the local minima. The annealing logic formation of SA technique is very easy though cooling criteria scheduling is pretty hard. SA is a single solution based optimization technique. Therefore the fundamental logic formation of SA technique mainly depends on random searching of good solution variables which will be accepted if objective function improves.

3.1 Steps of Simulated Annealing Algorithm

Step 1: Initially variables (x) are generated with the help of (3) where the uniform probability distribution function is used.

$$x_j^{(0)} = x_j^{\min} + \sigma_j(x_j^{\max} - x_j^{\min}) \tag{3}$$

where $j \in [1, \dots, D]$, D is the dimension of the system. With the initial solution vector objective function $f(x)$ is calculated.

Step 2: Initial temperature (T_i) and final temperature (T_f) is considered.

Step 3: According to the fundamental logic of SA technique geometric cooling schedule is set which subjects to $T=T_i$ and $T=\alpha*T$, where current temperature is stored at T . This way T will vary to crystallize or freeze till the final temperature comes. The multiplying factor α usually lies between 0.5 to 1.

Step 4: When the annealing process is started, a new set of solution is generated by (4) as given below.

$$x_{n+1} = x_n + randn \tag{4}$$

where, x_n is the initial solution, x_{n+1} is the new set of solution and ‘randn’ is the random number generation operator in MATLAB 7.1. Henceforth new set of objective function $f(x_{(n+1)})$ is calculated with x_{n+1} . With this the difference in energy level i.e., $\delta f = f(x_{(n+1)}) - f(x_n)$ is calculated to proceed the annealing schedule.

Step 5: Acceptance Criteria:

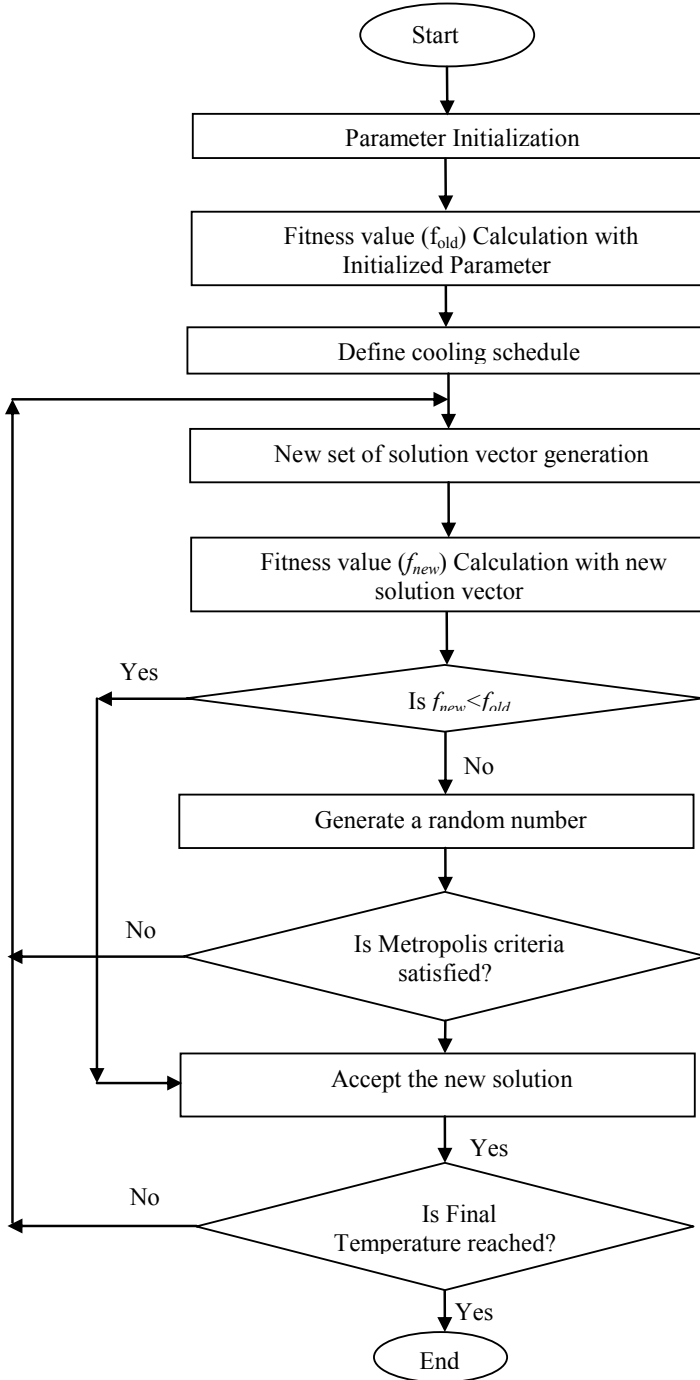


Fig. 1. Simulated Annealing Technique in Flow chart

If the new solution is better it will be accepted, otherwise a random number r will be generated. According to Metropolis criteria [8] new solution may be accepted if (5) satisfies.

$$p = \exp[-\delta f / kT] > r \quad (5)$$

where k is the Boltzmann's constant. If improved response is obtained, x_{n+1} and $f(x_{n+1})$ will be replaced by the previous data. The loop or the iteration will continue depending upon the terminating condition. Terminating condition may be set as reaching to the final temperature i.e. final iteration N ($=n: 1: N$) or arrival of certain number of repetitive result or even reaching to the optimal value. The working step of the SA method is explained in Fig. 1.

4 Result Analysis

4.1 SA Implementation with the RPD Problem

Initially a solution vector (x) is generated for the IEEE 118-bus system using Equ. (3). After considering T_i , T_f and α , cooling schedule is fixed. Then new solution vector is generated via Equ. (4). Subsequently new fitness function with the new solution vector is calculated. δf is calculated to proceed the annealing schedule. If the new solution is better it will be accepted, otherwise a random number r will be generated and Metropolis criteria i.e., Equ. (5) will be checked. The operating cycle will be continued until terminating condition is reached. In this paper the execution will terminate either reaching to the final temperature point or exceeding the preset maximum number of repetitive solution.

4.2 SA Based Parameter Selection

As programming software MATLAB 7.1 is considered here. In this paper a large power system network of IEEE 118-bus systems [16] is chosen for solving the real power loss minimization aspect. The said bus network consists of hundred and eighteen numbers of buses among them 69th bus is considered as slack bus having 1.07 pu as magnitude. In the system total fifty four number of generator buses is present. For solving the RPD problem the generator bus voltages are controlled. Remaining sixty three buses are load buses among them twelve buses are chosen for shunt compensation to minimize active power loss. Apart from this, nine tap changing transformers are placed in various positions of the network for the RPD problem solving issue. The bus data and line data are considered from the power system test case archive available online [16]. To calculate power loss Newton Raphson method is used as load flow study tool [10]. Optimal results of any evolutionary computation techniques depend on the proper choice of the different parameters. The chosen parameters for the proposed method are considered as $T_i=1$, $T_f=10^{-8}$, Maximum

repetition of result=150, $\alpha=0.9$. The considered range for the controlled variables (V_g as generator bus voltages, T as transformer tap setting and Q_c as shunt capacitors) and state variables (V_L as load bus voltages) are given in Table 1.

Table 1. Range of operation for the control variables

| Sl No. | Units | IEEE 118-Bus System |
|--------|-------|---|
| 1. | Pu | $0.95 \leq V_L \leq 1.10$ |
| 2. | Pu | $0.95 \leq V_g \leq 1.10$ |
| 3. | Pu | $0.90 \leq T \leq 1.10$ |
| 4. | Pu | $0.0 \leq Q_{c34} \leq 14.0$ |
| | | $0.0 \leq Q_{c44}, Q_{c45}, Q_{c46}, Q_{c83} \leq 10.0$ |
| | | $0.0 \leq Q_{c48} \leq 15.0$ |
| | | $0.0 \leq Q_{c74} \leq 12.0$ |
| | | $0.0 \leq Q_{c79}, Q_{c82}, Q_{c105} \leq 20.0$ |
| | | $0.0 \leq Q_{c107}, Q_{c110} \leq 6.0$ |

4.3 Result for IEEE 118-Bus System

By applying SA technique P_{LOSS} decreases to 128.735 MW and the % P_{LOSS} reduces to 2.10266 from the Newton Raphson based solution. In this case the average and the worst values are obtained as 130.173 MW and 131.405 MW respectively after fifty trial runs. Moreover standard deviation of the 50 trial run is calculated here within the paper via (6).

$$S_N = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (6)$$

Where (x_1, x_2, \dots, x_N) are the observed value of the sample items and \bar{x} is the mean value of the observation and N stands for the number of sample collected (50 trial runs in this case). The obtained standard deviation is 0.868405.

The Fig. 2 shows the best, average and worst response curves by the SA technique. According to the Fig. 2 the optimization process converges nearly at the final temperature whereas the other two observed responses converges even earlier.

The obtained results in terms of the controlled variables are given in Table 2. While analyzing the results one crucial point is to be mentioned that no result violates the stability and thermal operating limit. Even the results are obtained within 28.325 sec.

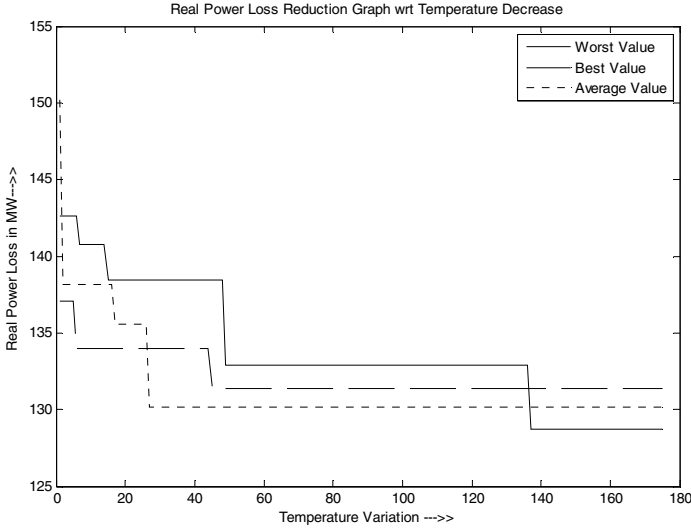


Fig. 2. SA Technique based result for IEEE 118-bus system

Table 2. Detailed variables for the SA Method

| Control Variables | Obtained Value | Control Variables | Obtained Value | Control Variables | Obtained Value |
|-------------------|----------------|-------------------|----------------|-------------------|----------------|
| V_{g1} | 1.0656 | V_{g61} | 1.0036 | V_{g111} | 0.9535 |
| V_{g4} | 1.0263 | V_{g62} | 1.0077 | V_{g112} | 0.9619 |
| V_{g6} | 1.0431 | V_{g65} | 1.0160 | V_{g113} | 1.0514 |
| V_{g8} | 1.0226 | V_{g66} | 0.9946 | V_{g116} | 0.9706 |
| V_{g10} | 1.0606 | V_{g70} | 1.0253 | T_{6-10} | 0.9008 |
| V_{g12} | 1.0972 | V_{g72} | 0.9809 | T_{4-12} | 1.0888 |
| V_{g15} | 1.0673 | V_{g73} | 0.9730 | T_{28-27} | 0.9902 |
| V_{g18} | 1.0235 | V_{g74} | 1.0587 | T_{6-10} | 0.9178 |
| V_{g19} | 0.9987 | V_{g76} | 1.0040 | T_{4-12} | 0.9594 |
| V_{g24} | 0.9743 | V_{g77} | 0.9606 | T_{28-27} | 1.0785 |
| V_{g25} | 1.0771 | V_{g80} | 0.9657 | T_{6-10} | 1.0873 |
| V_{g26} | 1.0496 | V_{g85} | 1.020 | T_{4-12} | 1.0053 |
| V_{g27} | 1.0459 | V_{g87} | 1.0516 | T_{28-27} | 1.0072 |
| V_{g31} | 1.0446 | V_{g89} | 1.0610 | Q_{c34} | 12.323 |
| V_{g32} | 1.0419 | V_{g90} | 1.0231 | Q_{c44} | 9.4884 |
| V_{g34} | 1.0819 | V_{g91} | 1.0484 | Q_{c45} | 7.6147 |
| V_{g36} | 0.9981 | V_{g92} | 1.0240 | Q_{c46} | 5.9607 |

Table 2. (continued.)

| | | | | | |
|-----------|--------|------------|--------|------------|--------|
| V_{g40} | 0.9503 | V_{g99} | 1.0309 | Q_{c48} | 12.842 |
| V_{g42} | 0.9841 | V_{g100} | 0.9635 | Q_{c74} | 1.0142 |
| V_{g46} | 0.9902 | V_{g102} | 1.0631 | Q_{c79} | 0.1484 |
| V_{g49} | 1.0177 | V_{g103} | 1.0733 | Q_{c82} | 3.1344 |
| V_{g54} | 0.9983 | V_{g104} | 1.0061 | Q_{c83} | 2.0100 |
| V_{g55} | 0.9612 | V_{g105} | 1.0222 | Q_{c105} | 2.6430 |
| V_{g56} | 0.9903 | V_{g107} | 0.9803 | Q_{c107} | 3.5365 |
| V_{g59} | 1.0807 | V_{g110} | 0.9630 | Q_{c110} | 4.8126 |

Table 3 shows the comparative analysis between the proposed methods and few other evolutionary techniques.

Table 3 shows a SA based comparative study with respect to different techniques including DE and PSO. From the table it is seen that SA is being able to reduce % P_{Loss} to 9.56, 8.0192, 7.867 and 0.064 with respect to L-SACP-DE, CGA, SPSO-07 and DE techniques respectively

Table 3. Comparative analysis

| Applied algorithms to RPD problem | Obtained minimized value of Power Loss in MW for the IEEE 118-bus system |
|--|--|
| L-SACP-DE [4] | 141.79864 |
| CGA [14] | 139.41498 |
| SPSO-07 [19] | 139.27522 |
| DE [13] | 128.318 |
| Proposed method in this paper i.e. SA | 128.235 |

Hence it can be concluded that the proposed method based result is very much impressive in respect to optimum value as well as the operating time.

5 Conclusion

In this paper a very significant problem of power system is analyzed by applying one of the vital meta-heuristic techniques. The single solution based technique SA shows improved result compared to the base case within very little operating time. Even the comparative study with few population based technique results the effectiveness of the proposed method. As a summary of the study it can be concluded that for the considered large bus system SA gives improved results in terms of both power loss as well as CPU operating time.

Acknowledgments. The authors of the paper are grateful to the Jadavpur University for providing a soothing research environment. The corresponding author is also grateful to the funding organization of her scholarship i.e. DST INSPIRE Fellowship programme.

References

1. Abido, M.A.: Optimal power flow using tabu search algorithm. *Electrical Power Components Systems* 30(5), 469–483 (2002)
2. Abou, E.E.A., Abido, M.A., Spea, A.R.: Optimal power flow using differential evolutionary algorithm. *Electrical Power Systems Research* 80, 878–885 (2010)
3. Bansilal, D.T., Parthasarathy, K.: Optimal reactive power dispatch algorithm for voltage stability improvement. *Electrical Power and Energy Systems* 18(70), 461–468 (1996)
4. Brest, J.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* 10(6), 646–657 (2006)
5. Liang, C.H., Chung, C.Y., Wong, K.P., Duan, X.Z., Tse, C.T.: Study of differential evolution for optimal reactive power flow. *IET Generation Transmission Distribution* 1(2), 253–260 (2007)
6. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. *Science New Series* 220(4598), 671–680 (1983)
7. Mahadevan, K., Kannan, P.S.: Comprehensive learning particle swarm optimization for reactive power dispatch. *Applied Soft Computing* 10, 641–652 (2010)
8. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics* 21(6), 1087–1092 (1953)
9. Osman, M.S., Abo-Sinna, M.A., Mousa, A.A.: A solution to the optimal power flow using genetic algorithm. *Applied Math Computation* 155(2), 391–405 (2004)
10. Sadat, H.: Power system analysis, ch. 6. Tata McGraw-Hill Publishing Company limited (2008) (fourteenth reprint)
11. Subbaraj, P., Rajnarayanan, P.N.: Optimal reactive power dispatch using self-adaptive real coded genetic algorithm. *Electrical Power Systems Research* 79, 374–381 (2009)
12. Talbi, E.G.: Metaheuristics from design to implementation, ch. 2, 3. John Wiley & Sons (2009)
13. Varadarajan, M., Swarup, K.S.: Differential evolution approach for optimal reactive power dispatch. *Electrical Power and Energy System* 30, 435–441 (2008)
14. Wu, Q.H., Cao, Y.J., Wen, J.Y.: Optimal reactive power dispatch using an adaptive genetic algorithm. *International Journal of Electrical Power Energy System* 20, 563–569 (1998)
15. Wu, Q.H., Ma, J.T.: Power system optimal reactive power dispatch using evolutionary programming. *IEEE Transactions on Power Systems* 10(3), 1243–1249 (1995)
16. Power system test case archive (December 2006), <http://www.wv.washington.edu/research/pstca>
17. Jwo, W.S., Liub, C.W.: Large-scale optimal VAR planning by hybrid simulated annealing/genetic algorithm. *Electrical Power and Energy Systems* 21, 39–44 (1999)
18. Roa-Sepulveda, C.A., Pavez-Lazo, B.J.: A solution to optimal power flow using simulated annealing. *Electrical Power and Energy Systems* 25, 47–57 (2003)
19. Standard PSO 2007 (SPSO-07) on the Particle Swarm Central, Programs section (2007), <http://www.particleswarm.info>

Hybrid Artificial Bee Colony Algorithm and Simulated Annealing Algorithm for Combined Economic and Emission Dispatch Including Valve Point Effect

Sundaram Arunachalam, R. Saranya, and N. Sangeetha

Department of E.E.E, St. Joseph's College of Engineering, Chennai
{mailto:arunachalam,sarane27,sangeethanatarajan55}@gmail.com

Abstract. For economic and efficient operation of power system optimal scheduling of generators in order to minimize fuel cost of generating units and its emission is a major consideration. This paper presents hybrid approach of using Artificial Bee Colony (ABC) and Simulated Annealing (SA) algorithm to solve highly constrained non-linear multi-objective Combined Economic and Emission Dispatch (CEED) having conflicting economic and emission objective. The mathematical formulation of multi objective CEED problem with valve point is formulated and then converted into single objective problem using price penalty factor approach. Performance of proposed hybrid algorithm is validated with IEEE 30 bus six generator systems and a 10 generating unit system. Programming is developed using MATLAB. The results obtained and computational time of proposed method is compared with ABC and SA algorithm. Numerical results indicates proposed algorithm is able to provide better solution with reasonable computational time.

Keywords: Economic Dispatch, Emission Dispatch, Power Loss, Multi objective Optimization.

1 Introduction

The Economic Dispatch problem can be stated as determining the least cost power generation schedule from a set of online generating units to satisfy the load demand at a given point of time [1]. Though the core objective of the problem is to minimize the operating cost satisfying the load demand, several types of physical and operational constraints make ED highly nonlinear constrained optimization problem, especially for larger systems [2]. In recent years, environmental considerations have regained considerable attention in the power system industry due to the significant amount of emissions like sulphur dioxide (SO_2) and nitrogen oxides (NO_x). So along with economic dispatch environmental dispatch must also be carried out. Since economic and emission objectives are conflicting in nature, a combined approach is the best to achieve an optimal solution [3].

Power plants commonly have multiple valves that are used to control the power output of the units. When steam admission valves in thermal units are first opened, a

sudden increase in losses is registered which results in ripples in the cost function. This effect is known as a valve point loading. Typically, the valve-point results in as each steam valve starts to open, the ripples like in to take account for the valve-point effects, sinusoidal functions are added to the quadratic cost function[4]. This type of problem is extremely difficult to solve with conventional gradient based techniques due to the abrupt changes and discontinuities present in the incremental cost function. The CEED problem with valve point effect is a Multi-objective problem which can be solved by Multi-objective Optimization [5].

Traditional methods such as lambda iteration, base point participation factor, gradient method and Newton method may not converge to feasible solutions for complex problems whose objective function is not continuously differentiable and has a discontinuous nature. This method fails for non-convex problem except dynamic programming in which no restriction is imposed on the shape of cost curves, also this method suffer from dimensionality problem and excessive computation effort.

The multi-objective optimization problem is formulated using Combined Economic Emission Dispatch (CEED) approach which merges the cost and emission objectives into one optimization function such that equal importance is assigned to both objectives [6]. Several classical methods were proposed to solve economic dispatch problem such as lambda iteration method, Gradient method, base point and participation factor method [7]. Heuristic method for solving economic dispatch is presented in [8-20]. The formulation of multi objective CEED problem by weighting function method and priority ranking method using hybrid ABC-PSO method is presented in [21].

In this paper the CEED problem is first solved by ABC method and the optimal schedules are obtained. This schedule is given as starting point for Simulated Annealing and the optimal schedules are obtained. This approach combines the advantages of faster convergence of ABC method and robustness of SA method to find the global solution of highly non linear CEED problem with valve point effect.

The proposed method converts the multi –objective problem into a single objective problem by using a penalty factor approach. Methods which convert multi objective into single objective using weights generate the non dominated solution by varying the weights, thus requiring multiple runs to generate the desired Pareto set of solution. Various other approaches given in [23],[24] and [25] solves the conflicting objective functions simultaneously using multi objective evolutionary search strategies and find out compromise solution.

2 Formulation of Combined Economic and Emission Dispatch (CEED)

The multi-objective CEED problem is formulated by combining the economic dispatch problem and emission dispatch problem into a single objective using price penalty factor method.

2.1 Formulation of Multi Objective CEED problem

The objective of the multi-objective CEED problem which has two conflicting objectives as economic and emission objective is to find the optimal schedules of the thermal generating units which minimizes the total fuel cost and emission from the thermal units subject to power balance equality constraint and bounds. The mathematical formulation of the multi objective CEED problem is given below

$$\min [F_{TV}, E_T] \quad (1)$$

subject power balance equation given in (2) and bounds given in (3)

$$\sum_{i=1}^{nb} P_i - P_D - P_L = 0 \quad (2)$$

$$P_{i,min} \leq P_i \leq P_{i,max} \quad (3)$$

where

F_{TV} Total fuel cost of Ng generating units with valve point effect

$$F_{TV} = \sum_{i=1}^{Ng} F(P_i) = \sum_{i=1}^{Ng} \alpha_i P_i^2 + b_i P_i + c_i + d_i * \sin(e_i(P_{i,min} - P_i)) \$/h \quad (4)$$

E_T Total emission cost Ng generating units

$$E_T = \sum_{i=1}^{Ng} E(P_i) = \sum_{i=1}^{Ng} \alpha_i P_i^2 + \beta_i P_i + \gamma_i + \eta_i e^{\delta_i P_i} Kg/h \quad (5)$$

$\alpha_i, \beta_i, \gamma_i, \eta_i, \delta_i$ Emission coefficients of thermal unit i

a_i, b_i, c_i Fuel cost coefficients of thermal unit i

e_i, f_i Coefficients to model the effect of valve point of thermal unit i

Ng Total number of thermal generating units

nb Number of buses

P_i Power generation of thermal unit i

P_D Total demand of the system

P_L Real Power transmission loss in the system

$P_{i,min}$ Minimum generation limit of thermal unit i

$P_{i,max}$ Maximum generation limit of thermal unit i

In the above formulation the transmission loss in the system is calculated using B matrix coefficients calculated from load flow solution as given in [14] and incorporated into power balance equality constraint. These loss coefficients are independent of slack bus. The transmission loss in the system is expressed using B matrix coefficients as

$$P_L = \sum_{i=1}^{nb} \sum_{j=1}^{nb} P_i B_{ij} P_j + \sum_{i=1}^{nb} B_{i0} P_i + B_{00} \quad (6)$$

The above multi objective problem can be combined into a single objective problem using price penalty factor approach. The price penalty factor approach to combine this multi objective problem in to a single objective is given in the next section.

2.2 Penalty Factor Approach

As mentioned earlier Multi-objective CEED is converted into a single objective problem using penalty factor approach. The sequential steps involved in calculating penalty factor are listed below

- Evaluate the maximum cost of each generator at its maximum output.

$$F(P_{i,max}) = a_i P_{i,max}^2 + b_i P_{i,max} + c_i + e_i * \sin(f_i(P_{i,min} - P_{i,max})) \$/h \quad (7)$$

- Evaluate the maximum emission of each generator at its maximum output.

$$E(P_{i,max}) = \sum_{i=1}^{Ng} \alpha_i P_{i,max}^2 + \beta_i P_{i,max} + \gamma_i + \eta_i e^{\delta_i P_{i,max}} Kg/h \quad (8)$$

- Divide the maximum cost of each generator by its maximum emission

$$h_i = \frac{F(P_{i,max})}{E(P_{i,max})} \quad (9)$$

- Arrange h_i in ascending order. Add $P_{i,max}$ of each unit one at a time starting from the smallest h_i unit until it meets the total demand P_D
- At this stage, h_i associated with the last unit in the process is the price penalty factor h in $\$/Kg$ for the given load.

2.3 Conversion of Multi Objective CEED to Single Objective Formulation

The multi objective CEED is converted into single objective optimization using price penalty factor and the respective formulation is given below

$$\min \sum_{i=1}^{Ng} F_{TV}(P_i) + h \sum_{i=1}^{Ng} E(P_i) \quad (10)$$

subject to power balance equality constraint and bounds given below

$$\sum_{i=1}^{nb} P_i - P_D - P_L = 0 \quad (11)$$

$$P_{i,min} \leq P_i \leq P_{i,max} \quad (12)$$

In (10) the $F_{TV}(P_i)$ can also be replaced by $F_T(P_i)$ if the valve point effect has to be neglected. $F_T = \sum_{i=1}^{Ng} F(P_i) = \sum_{i=1}^{Ng} a_i P_i^2 + b_i P_i + c_i$. In this paper the above formulation is solved using hybrid ABC-SA method. A Brief algorithm of ABC and SA is presented in the next section

3 Artificial Bee Colony Algorithm

ABC algorithm is one of the most promising methods for solving complex non-smooth optimization problems in power systems. It simulates the behavior of real bees for solving optimization problems. The colony of artificial bees consists of three

groups of bees: employed bees, onlookers and scouts. The first half of the colony consists of the employed artificial bees and the second half includes the onlookers. The number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source has been exhausted by the bees becomes a scout. Communication among bees related to the quality of food sources takes place in the dancing area. This dance is called a waggle dance. The four control parameters of ABC algorithm are

- NP – The number of colony size(employee bee+ onlookers)
- $Food\ number = \frac{NP}{2}$ The number of food sources equals the half of the colony size.
- $Limit$ – A food source which could not be improved through limit trials is abandoned by its employee bee.
- $Max\ cycle$ –The number of cycles for foraging (a stopping criterion).

4 Simulated Annealing

Simulated Annealing (SA) is a stochastic optimization technique and it can be used to solve our CEED problem since our objective is to find an optimal solution. Simulated Annealing is a random search technique which exploits an analogy between the way in which a metal cools and freezes into a minimum energy perfect crystalline structure with minimum defects (the annealing process) and the search for a minimum in a more general system. There are four control parameters that are directly associated with its convergence (to an optimized solution) and its efficiency. They are Initial temperature, Final temperature, Rate of temperature decrement, Iteration at each temperature.

5 Hybrid ABC-SA Algorithm

In this paper, a hybrid ABC-SA algorithm is proposed for solving CEED problem. The proposed ABC-SA is a method of combining the advantages of faster computation of Artificial Bee colony Algorithm with robustness of Simulated Annealing (SA) so as to increase the global search capability. The ABC algorithm starts with a set of solutions and based upon the survival of fittest principle, only the best solution moves from one phase to another. This process is repeated until the any of the convergence criteria is met. At the end of the iterations the optimal solution is the one with the minimum total cost out of the set of solutions. The time of convergence of ABC depends upon the values of the randomly set control parameters. SA algorithm starts with an initial operating solution and every iteration improves the solution until the convergence criteria is met. The optimal solution obtained from SA algorithm depends upon the quality of the initial solution provided. In this paper the initial solution provided to SA is the optimal solution obtained from ABC algorithm. Since a best initial solution is given to SA algorithm the optimal solution obtained from this Hybrid approach is better than the solution obtained from ABC or SA algorithms.

The sequential steps involved in the proposed algorithm ABC-SA algorithm is given below

1. The cost data, emission data and valve point data of each generating unit are read and system load is also specified.
2. The minimum and maximum operating limits of the generating units are specified.
3. The penalty factor to combine the multi objective problem into a single objective problem is obtained from the algorithm given in section 2.2.
4. Using this penalty factor a lossless dispatch is carried out using ABC algorithm for the formulation given by equation (10) to (12) with $P_L = 0$.
5. With the solution obtained an AC power flow is carried out and the B-loss coefficients are obtained [14]. These coefficients are used for calculation of real power loss in the subsequent iterations.
6. The various control parameters of the ABC algorithm are initialized. Formulation given by equation (10) to (12) is solved using the ABC algorithm developed in MATLAB.
7. ABC runs till its stopping criterion (the maximum number of iterations) is met,
8. In order to obtain the optimal control parameters, the steps 7 to 10 is run many times with one control parameter fixed and all other control parameters are varied. This step is repeated to find the best control parameter for ABC algorithm.
9. With the best control parameters set, the ABC algorithm is carried and the optimal solution is obtained. With this optimal schedule an AC load flow is carried out and using the solutions of AC load flow the new B Coefficients are obtained and considered for the subsequent iteration.
10. The optimal solution of ABC is given as the starting point (Initial guess vector) to the SA algorithm and the control parameters of SA are set.
11. Then, the SA algorithm starts its search process and it is run until its stopping criterion is met.
12. With this optimal solution the total fuel cost of the thermal generating units and its emission cost are calculated.

6 CASE STUDY – IEEE 30 Bus System

In order to validate the proposed algorithm a case study with IEEE 30 bus test system consisting of 6 generating units and 41 transmission lines is carried out. The valve point effect is not considered. The load demand of this test system is $500 MW$. The total load in the IEEE 30 bus system is $283.4 MW$. Each of the real power demand is scaled to increase the total demand to $500 MW$. The fuel cost coefficients and emission function coefficients to minimize sulphur oxides(SOx) and Nitrogen oxides(NOx) caused by thermal plant along with generator capacity limits of each generator are given in appendix table A1. With the schedules obtained from lossless dispatch a load flow is carried out and the B-co-efficient matrix computed from the load flow analysis is given in appendix table A2. The penalty factor to combine the multi objective problem into single objective problem is calculated as given in section 2.2 and is found to be $h = 43.15 \$/kg$

For this system the optimal dispatch is obtained using ABC, SA and Hybrid ABC-SA algorithm and are compared in the subsequent sections

6.1 Solution of CEED Problem Using ABC Method

Since evolutionary algorithm is used to solve CEED, certain parameters of the algorithm have to be randomly adjusted. The optimal control parameters for ABC algorithm are found by varying each parameter and setting one parameter constant at a time and this process is repeated to find the optimal control parameters. The optimal control parameter for this test system is found as $NP = 30$, $limit = 300$, $maxcycle = 500$. With these parameters ABC algorithm is run for twenty times and the schedules are shown in table 1. The optimal schedules from the ABC algorithm is shown in bold in table 1.

Table 1. Optimal Schedules of CEED problem obtained using ABC algorithm

| RUNS | P1 | P2 | P3 | P4 | P5 | P6 | LOSS | Fuel | Emission | Total | TIME |
|------|--------------|--------------|---------------|--------------|---------------|---------------|-------------|--------------|---------------|--------------|------------|
| | (MW) | (MW) | (MW) | (MW) | (MW) | (MW) | (MW) | COST | (kg/hr) | COST | (sec) |
| | | | | | | | | (\$/hr) | | (\$/hr) | |
| 1 | 10.02 | 23.96 | 94.56 | 119.11 | 134.22 | 127.99 | 9.89 | 27671 | 273.91 | 39491 | 5.50 |
| 2 | 10.01 | 18.40 | 108.26 | 99.22 | 131.72 | 141.52 | 9.17 | 27583 | 271.90 | 39316 | 5.9 |
| 3 | 10.00 | 12.13 | 109.93 | 89.08 | 130.91 | 157.46 | 9.53 | 27556 | 276.34 | 39480 | 5.90 |
| 4 | 10.03 | 10.00 | 96.851 | 104.87 | 161.80 | 125.71 | 9.27 | 27499 | 277.11 | 39457 | 5.65 |
| 5 | 10.06 | 11.85 | 94.82 | 72.35 | 194.52 | 125.79 | 9.41 | 27485 | 291.81 | 40076 | 5.63 |
| 6 | 10.00 | 10.67 | 125.98 | 84.97 | 138.11 | 138.17 | 7.93 | 27519 | 279.03 | 39559 | 3.49 |
| 7 | 10.01 | 11.62 | 91.69 | 78.40 | 191.13 | 126.76 | 9.63 | 27484 | 288.68 | 39941 | 3.43 |
| 8 | 10.01 | 10.07 | 118.68 | 108.28 | 130.46 | 130.64 | 8.17 | 27552 | 277.58 | 39530 | 3.51 |
| 9 | 10.03 | 10.46 | 116.08 | 65.82 | 171.25 | 134.61 | 8.27 | 27473 | 286.37 | 39830 | 3.43 |
| 10 | 10.02 | 10.00 | 120.56 | 74.49 | 163.23 | 129.56 | 7.88 | 27471 | 282.72 | 39670 | 3.49 |
| 11 | 10.02 | 10.30 | 93.12 | 59.36 | 202.65 | 134.30 | 9.77 | 27502 | 301.87 | 40528 | 3.47 |
| 12 | 10.00 | 10.03 | 111.93 | 102.75 | 147.25 | 126.35 | 8.33 | 27504 | 275.67 | 39400 | 3.42 |
| 13 | 10.02 | 11.70 | 122.89 | 35.00 | 186.99 | 141.78 | 8.39 | 27563 | 313.29 | 41082 | 3.46 |
| 14 | 10.15 | 10.13 | 108.59 | 106.17 | 140.35 | 133.37 | 8.79 | 27525 | 274.78 | 39382 | 3.42 |
| 15 | 10.04 | 10.02 | 40.08 | 88.23 | 200.25 | 167.03 | 15.68 | 27728 | 312.28 | 41592 | 3.45 |
| 16 | 10.06 | 10.99 | 143.16 | 77.66 | 138.85 | 126.19 | 6.93 | 27553 | 289.35 | 40039 | 3.46 |
| 17 | 10.00 | 10.41 | 123.95 | 95.43 | 130.03 | 138.23 | 8.08 | 27541 | 277.93 | 39534 | 3.43 |
| 18 | 10.13 | 35.81 | 88.20 | 90.35 | 158.36 | 127.65 | 10.53 | 27704 | 270.87 | 39392 | 3.50 |
| 19 | 10.02 | 14.65 | 97.57 | 121.73 | 140.04 | 125.40 | 9.44 | 27598 | 277.23 | 39560 | 3.45 |
| 20 | 10.00 | 10.07 | 115.74 | 76.17 | 158.14 | 138.24 | 8.37 | 27474 | 279.73 | 39545 | 3.48 |

At the end of several trial runs the best optimal fuel cost is found to be 27583 \$/hr and the emission is found to be 271.90 kg/hr. The total cost of the system is obtained as 39316 \$/kg. These results are obtained within a computation time of 5.65 seconds.

6.2 Solution of CEED Problem Using SA Method

Similar to ABC method, the parameters of SA method is also tuned to certain trial values like (i.e. cool schedule=0.5T and 0.8T, Temp (T) =200 and 300, Max tries=8000 and 10000). Among the several results the best optimal solution is obtained with cool schedule=0.8T, T=100 and Max tries=8000 and is shown in table 2.

At the end of several trial runs the best optimal fuel cost is found to be 27498 \$/hr and the emission is found to be 275.60 kg/hr. The total cost of the system is obtained as 39391 \$/kg. These results are obtained within a computation time of 58.44 seconds.

6.3 Solution of CEED Problem Using Hybrid ABC-SA Method

In this method the best schedule obtained in ABC method is given as initial start to SA algorithm and the parameters of the SA are set as T= 100, cool schedule= 0.8T and Max tries= 5000. The optimal schedule obtained from the hybrid method is shown in table 2. The optimal fuel cost is found to be 27588 \$/hr and the emission is found to be 271.71 kg/hr. The total cost of the system is obtained as 39313 \$/kg which is better than the ABC method and the SA method as shown by the comparison table 2. These results are obtained within a computation time of 18.57 seconds. The computational time is more than ABC method but the optimal cost is further reduced than the ABC method. The results are better than the hybrid approach used in [22] and this is mainly due to the computation of loss using B loss coefficients obtained from lossless dispatch in hybrid ABC-SA method which is very reasonable when compared to the B loss matrix used in [22].

Table 2. Comparison of the optimal schedules obtained by ABC, SA and Hybrid ABC-SA method and Hybrid ABC-PSO method used in [22]

| Optimal Schedules | ABC | SA | ABC-SA | Ref [22] |
|--------------------|------------|--------|--------------|----------|
| P1(MW) | 10.01 | 10.00 | 10.90 | 54.6 |
| P2(MW) | 18.40 | 10.00 | 18.37 | 32.484 |
| P3(MW) | 108.26 | 94.53 | 108.41 | 48.548 |
| P4(MW) | 99.22 | 93.12 | 99.29 | 77.517 |
| P5(MW) | 131.72 | 155.81 | 131.68 | 167.28 |
| P6(MW) | 141.52 | 146.49 | 140.62 | 137.29 |
| LOSS(MW) | 9.17 | 9.97 | 9.31 | 17.718 |
| FUEL OST(\$/hr) | 27583 | 27498 | 27588 | 28157 |
| EMISSION (lb/hr) | 271.90 | 275.60 | 271.71 | 288.01 |
| TOTAL COST (\$/hr) | 39316 | 39391 | 39313 | 40584.6 |
| TIME (sec) | 5.9 | 58.44 | 18.57 | - |

6.4 Case Study on 10 Generating Units with Valve Point Effect

This case study consists of a standard test system with 10 generating units. The valve point effect is considered. The complexity to the solution process has significantly increased. In as much as this is a larger system with higher non-linearity, it has more

local minima and thus it is difficult to attain the global solution. The load demand of this test system is 2000 MW. The fuel cost coefficients with valve point co-efficient and emission function coefficients to minimize sulphur oxides(SOx) and Nitrogen oxides(NOx) caused by thermal plant along with generator capacity limits of each generator are given in appendix table A3& table A4. Here the losses in the system are also considered. The B matrix of the test system is tabulated in appendix table A5. As mentioned earlier economic and emission objectives are combined using Penalty factor approach. The penalty factor obtained from the procedure described in section 2.2. is $h = 51.99\$/kg$

In this method the best optimal schedule obtained in ABC method is given as initial start to SA algorithm and the SA method. The best optimal is obtained at T= 100, cool schedule= 0.8T and Max tries= 8000. The optimal power schedule of the 10 generating units and the loss of the system is tabulated in table 3. At the end of several trial runs the best optimal fuel cost is found to be 113510 \$/hr and the emission is found to be 4169 kg/hr. The total cost of the system is obtained as 330210 \$/kg. These results are obtained with a computation time of 22.35 seconds.

Table 3. Optimal Schedules obtained using Hybrid ABC-SA Method

| P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | LOSS |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| (MW) | (MW) | (MW) | (MW) | (MW) | (MW) | (MW) | (MW) | (MW) | (MW) | (MW) |
| 55.00 | 70.32 | 81.18 | 96.47 | 159.72 | 155.92 | 229.31 | 337.57 | 431.34 | 467.57 | 84.45 |

7 Conclusion

This paper has implemented a hybrid ABC and SA algorithm for solving the combined economic and emission dispatch problem including valve point effect. Results obtained from the proposed method are compared with ABC, SA and Hybrid ABC-PSO method. From the case studies carried out on the test systems and the results obtained indicate the proposed algorithm is able to find better optimal schedules in a reasonable computational time.

References

1. Wood, A.J., Wollenberg, B.F.: Power Generation, Operation and control. John Wiley and Sons, New York (1996)
2. Jeyakumar, A.E., Victorie, T.A.A.: Hybrid PSO-SQP for economic dispatch with valve-point effect. *Electr. Power Syst. Res.* 71(1), 51–59 (2004)
3. Wang, W.S., Chen, Y.M.: A Particle Swarm approach to solve environmental/economic dispatch problem. *International Journal of Engineering Computations* 1, 157–172 (2010)
4. Chiang, C.-L.: Improved Genetic Algorithm for Power Economic Dispatch of units with Valve-Point Effects and Multi Fuels. *IEEE Transactions on Power Systems* 20(4) (2005)

5. Hemamalini, S., Simon, S.P.: Economic/Emission Load Dispatch using Artificial Bee colony Algorithm. In: Proc. of Int. Conf. on Control, Communication and Power Engineering (2010)
6. Dinu, S., Odagescu, L., Moise, M.: Environmental Economic Dispatch Optimization using a Modified Genetic Algorithm. *International Journal of Computer Applications* (0975-8887) 20(2) (2011)
7. Saadat, H.: *Power System Analysis*. TATA MCGraw Hill Edition
8. Altun, H., Uzam, M., Yalcinoz, T.: Economic Dispatch Solution using a Genetic Algorithm based on Arithmetic Crossover. In: IEEE Porto Power Tech. Conference, September 10-12, Porto, Portugal (2001)
9. Senthil, K.: Combined Economic and Emission Dispatch using Evolutionary Programming Techniques. *IJCA Special Issue on "Evolutionary Computation for Optimization Techniques" ECOT* (2010)
10. Niu, Q., Wang, X., Zhou, Z.: An efficient cultural Particle Swarm Optimization for Economic Load Dispatch with Valve-Point Effect. *Sciverse Science Direct, Procedia Engineering* 23, 828–834 (2011)
11. Manikandan, K., Senthil, K.: Economic Thermal Power Dispatch with Emission constraint and Valve Point Effect Loading using Improved Tabu Search Algorithm. *IJCA* (0975-8887) 3(9) (2010)
12. Coelho, L.D.S., Mariani, V.C.: An Improved Harmony Search Algorithm for Power Economic Load Dispatch. *Science Direct, Energy Conversion and Management* 50, 2522–2526 (2009)
13. Hassan, E.E., Rahman, T.K.A., Zakria, Z.: Improved Adaptive Tumbling Bacterial Foraging Algorithm for Emission constrained Economic Dispatch Problem. In: *Proceedings of the World Congress on Engineering*, vol. 2 (2012)
14. John, J., Grainer William, D., Stevenson, J.: *Power system Analysis*. TATA McGraw-Hill Edition
15. Karaboga, D.: AN Idea Based On Honey Bee Swarm for Numerical Optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
16. Dixit, G.P., Dubey, H.M., Pandit, M., Panigrahi, B.K.: Economic Load Dispatch using Artificial Bee Colony Optimization. *International Journal of Advance in Electronics Engineering* 1, ISSN 2278-215X
17. Hemamalini, S., Simon, S.P.: Economic Load Dispatch With Valve Point Effect Using Artificial Bee Colony algorithm. In: XXXII National Systems Conference, NSC, December 17-19 (2008)
18. Ramaswamy, M., Sasikala: Optimal based Economic Emission Dispatch using Simulated Annealing. *International Journal of Computer Applications*(0975-8887) 1(10)
19. Dubey, H.M., Vishwakarma, K.K., Pandit, M., Panigrahi, B.K.: Simulated annealing approach for solving economic load dispatch problems with valve point loading effects. *International Journal of Engineering, Science and Technology* 4(4), 60–72 (2012)
20. Vlachos, A.: Simulated Annealing Algorithm for Environmental/Economic Dispatch Problem. *Journal of Information and Computing Science* 6(1), 64–72 (2011) ISSN 1746-7659
21. Manteau, E.D., Odero, N.A.: Combined economic emission dispatch using hybrid ABC-PSO algorithm with valve point loading effect. *International Journal of Scientific and Research Publications* 2(12) (December 2012) ISSN 2250-3153
22. Manteau, E.D., Odero, N.A.: Multi Objective Environmental/economic Dispatch Solution using ABC_PSO Hybrid Algorithm. *International Journal of Scientific and Research Publications* 2(12) (December 2012) ISSN 2250-3153

23. Panigrahi, B.K., Ravikumar Pandi, V., Das, S., Das, S.: Multi objective fuzzy dominance based bacterial foraging algorithm to solve economic emission dispatch problem. In: The 3rd International Conference on Sustainable Energy and Environmental Protection, SEEP 2009, vol. 35(12), pp. 4761–4770 (December 2010)
24. Panigrahi, B.K., Ravikumar Pandi, V., Sharma, R., Das, S., Das, S.: Multi objective bacterial foraging algorithm for electrical load dispatch problem. Energy Conservation and Management 52(2), 1334–1342 (2011)
25. Agrawal, S., Panigrahi, B.K., Tiwari, M.: Multi objective particle swarm algorithm with fuzzy clustering for electrical power dispatch. IEEE Transaction on Evolutionary Computation 12(5), 529–541 (2008)

Table A1. Fuel and Emission Coefficients for IEEE 30 Bus System with 6 Generator Bus

| UNIT | a (\$/hr) | b (\$/MWhr) | c (\$/(MW) ² hr) | α (kg/MW) ² hr | β (kg/MWhr) | γ (kg/hr) | P _{Max} (MW) | P _{Min} (MW) |
|------|-----------|-------------|-----------------------------|---------------------------|-------------|-----------|-----------------------|-----------------------|
| 1 | 0.15247 | 38.53973 | 756.79886 | 0.00419 | 0.32767 | 13.85932 | 125 | 10 |
| 2 | 0.10587 | 46.15916 | 451.32513 | 0.00419 | 0.32767 | 13.85932 | 150 | 10 |
| 3 | 0.02803 | 40.39655 | 1049.32513 | 0.00683 | -0.54551 | 40.2669 | 250 | 40 |
| 4 | 0.03546 | 38.30553 | 1243.5311 | 0.00683 | -0.54551 | 40.2667 | 210 | 35 |
| 5 | 0.02111 | 36.32782 | 1658.5696 | 0.00461 | -0.51116 | 42.89553 | 325 | 130 |
| 6 | 0.01799 | 38.2704 | 13356.2704 | 0.00461 | -0.51116 | 42.89553 | 315 | 125 |

Table A2. B-Loss Coefficient Matrix for IEEE 30 BUS System

| | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|
| 0.8422 | 0.1411 | -0.0069 | 0.0033 | 0.0009 | 0.0012 | 0.0304 |
| 0.0009 | 0.0019 | 0.1411 | 0.0677 | -0.0023 | 0.0001 | 0.0077 |
| -0.0069 | -0.0023 | 0.0167 | -0.0087 | -0.0069 | 0.0069 | -0.0009 |
| 0.0055 | 0.0037 | -0.0033 | 0.0001 | -0.0087 | 0.0164 | 0.0012 |
| 0.0009 | 0.0009 | -0.0069 | 0.0055 | 0.0121 | 0.0005 | -0.0001 |
| 0.0005 | 0.0258 | 0.0012 | 0.0019 | -0.0069 | 0.0037 | -0.0012 |
| 0.0304 | 0.0077 | -0.0009 | 0.0012 | -0.0001 | -0.0012 | 0.0014 |

Table A3. Fuel Cost Coefficients of 10 Generating Units

| UNIT | a (\$/MW) ² hr | b (\$/(MW) ² hr) | c (\$/(MW) ² hr) | d (\$/hr) | e rad/MW |
|------|---------------------------|-----------------------------|-----------------------------|-----------|----------|
| 1 | 0.12951 | 40.5407 | 1000.40 | 33 | 0.0174 |
| 2 | 0.10908 | 39.5804 | 950.606 | 25 | 0.0178 |
| 3 | 0.12511 | 36.5104 | 900.705 | 32 | 0.0162 |
| 4 | 0.12111 | 39.5104 | 800.705 | 30 | 0.0168 |
| 5 | 0.15247 | 38.539 | 756.799 | 30 | 0.0148 |
| 6 | 0.10587 | 46.1592 | 451.325 | 20 | 0.0163 |
| 7 | 0.03546 | 38.3055 | 1243.53 | 20 | 0.0152 |
| 8 | 0.02803 | 40.3965 | 1049.99 | 30 | 0.0128 |
| 9 | 0.02111 | 36.3278 | 1658.56 | 60 | 0.0136 |
| 10 | 0.01799 | 38.2704 | 1356.65 | 40 | 0.0141 |

Table A4. Emission Coefficients of 10 Generating Units

| α (lb/MW) ² hr | β (lb/MWhr) | γ lb/hr | eta lb/hr | Lambda (1/MW) | P_{Max} (MW) | P_{Min} (MW) |
|-------------------------------------|-------------------|-------------------|--------------|------------------|-------------------|-------------------|
| 0.04702 | -3.9864 | 360.0012 | 0.25475 | 0.01234 | 55 | 10 |
| 0.04652 | -3.9524 | 350.0012 | 0.25473 | 0.01234 | 80 | 20 |
| 0.04652 | -3.9023 | 330.0056 | 0.25163 | 0.01215 | 120 | 47 |
| 0.04652 | -3.9023 | 330.0056 | 0.25163 | 0.01215 | 130 | 20 |
| 0.0042 | 0.3277 | 13.8593 | 0.2497 | 0.012 | 160 | 50 |
| 0.0042 | 0.3277 | 13.8593 | 0.2497 | 0.012 | 240 | 70 |
| 0.0068 | -0.5455 | 40.2699 | 0.248 | 0.0129 | 300 | 60 |
| 0.0068 | -0.5455 | 40.2699 | 0.2499 | 0.01203 | 340 | 70 |
| 0.0046 | -0.5112 | 42.8955 | 0.2547 | 0.01234 | 470 | 135 |
| 0.0046 | -0.5112 | 42.8955 | 0.2547 | 0.01234 | 470 | 150 |

Table A5. B Loss Coefficients of 10 Generating Units

| | | | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0.000049 | 0.000014 | 0.000015 | 0.000016 | 0.000016 | 0.000016 | 0.000016 | 0.000016 | 0.000016 | 0.000016 |
| 0.000014 | 0.000045 | 0.000016 | 0.000016 | 0.000017 | 0.000015 | 0.000015 | 0.000016 | 0.000018 | 0.000018 |
| 0.000015 | 0.000016 | 0.000039 | 0.00001 | 0.000012 | 0.000012 | 0.000014 | 0.000014 | 0.000016 | 0.000016 |
| 0.000015 | 0.000016 | 0.00001 | 0.00004 | 0.000014 | 0.00001 | 0.000011 | 0.000012 | 0.000014 | 0.000015 |
| 0.000016 | 0.000017 | 0.000012 | 0.000014 | 0.000035 | 0.000011 | 0.000013 | 0.000013 | 0.000015 | 0.000016 |
| 0.000017 | 0.000015 | 0.000012 | 0.00001 | 0.000011 | 0.000036 | 0.000012 | 0.000012 | 0.000014 | 0.000015 |
| 0.000017 | 0.000015 | 0.000012 | 0.00001 | 0.000011 | 0.000036 | 0.000012 | 0.000012 | 0.000014 | 0.000015 |
| 0.000017 | 0.000015 | 0.000012 | 0.00001 | 0.000011 | 0.000036 | 0.000012 | 0.000012 | 0.000014 | 0.000015 |
| 0.000018 | 0.000016 | 0.000014 | 0.000012 | 0.000013 | 0.000012 | 0.000016 | 0.00004 | 0.000015 | 0.000016 |
| 0.000018 | 0.000016 | 0.000014 | 0.000012 | 0.000013 | 0.000012 | 0.000016 | 0.00004 | 0.000015 | 0.000016 |
| 0.000019 | 0.000018 | 0.000016 | 0.000014 | 0.000015 | 0.000014 | 0.000016 | 0.000015 | 0.000042 | 0.000019 |
| 0.00002 | 0.000018 | 0.000016 | 0.000015 | 0.000016 | 0.000015 | 0.000018 | 0.000016 | 0.000019 | 0.000044 |

Spectrum Allocation in Cognitive Radio Networks Using Firefly Algorithm

Kiran Kumar Anumandla, Shravan Kudikala,
Bharadwaj Akella Venkata, and Samrat L. Sabat

School of Physics, University of Hyderabad, Hyderabad -500046, India
kirankumara@uohyd.ac.in, slssp@uohyd.ernet.in

Abstract. In cognitive radio network, Spectrum Allocation (SA) problem is a NP-hard problem which needs to be solved in real time. In this work, a recent bio-inspired heuristic Firefly algorithm (FA) is used for solving SA problem. Three objective functions namely (a) Max-Sum-Reward (*MSR*), (b) Max-Min-Reward (*MMR*) and (c) Max-Proportional-Fair (*MPF*) are optimized to maximize the network capacity. The performance of FA is compared with Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) algorithms in terms of quality of solution (network capacity) and timing complexity. The simulation result reveals that the Firefly algorithm improved quality of solution and timing complexity by 17% and 100% respectively compared to PSO, in contrast to 13% and 103% compared to ABC algorithm. FA proved to give maximum utilization of network capacity by assigning conflict free channels to secondary users.

1 Introduction

Now a days, the numbers of wireless devices have increased which in turn resulted in increased demand for radio spectrum. To eliminate the interference between the spectrum users, current policies allocate fixed spectrum slice to each wireless application. Due to the fixed licensing policy only 6% of spectrum is utilized temporally and spatially [15]. Studies have reported that the under-utilized spectrum can be utilized using cognitive radio technology. In this technology, each cognitive radio user (Secondary user) can adapt to various technologies and utilize the vacant spectrum with out any interference to the licensed users (Primary users). Cognitive radio is built on a software-defined radio and work as an intelligent wireless communication system, that can sense, learn and adapt to statistical variations in the input and utilize vacant spectrum efficiently [7]. By using four main functionalities such as spectrum sensing, spectrum management, spectrum mobility and spectrum sharing, secondary users can opportunistically utilize the spectrum holes. Secondary users must opportunistically utilize the vacant licensed spectrum subject to the constraints imposed by licensed users. Due to spectrum heterogeneity, spectrum available to cognitive users vary with respect to location and time, because of traffic variation of primary users. The interference between the neighboring users can be avoided by proper coordination between them. This in turn will improve the spectrum utilization. The main objective of the cognitive radio is to maximize the spectrum utilization. This demands dynamic spectrum access, hence an efficient spectrum allocation algorithm need to be developed to support dynamic spectrum access and to provide fairness across all the secondary users.

The spectrum allocation model has been implemented by using game theory [16], auction and pricing mechanisms [8], graph coloring [29] and local bargaining algorithms [14]. This model can also be formulated as an optimization problem and it has been solved by using different evolutionary algorithms like genetic algorithm (GA), quantum genetic algorithm (QGA), particle swarm optimization (PSO) [28] and Artificial Bee Colony (ABC) [22].

ABC algorithm is a recent swarm intelligent optimization algorithm inspired by the foraging behavior of honey bees [9,10,11]. The colony consists of three groups of bee: employed bees, onlookers and scouts. Each employed bee is associated with a food source and onlooker bees contribute the information of the food sources found by employed bees to choose better one. If some food sources are not improved for more cycles, the scouts are translated to a few employed bees which search for new food sources. PSO is another general purpose swarm based optimization method [20,12,13], inspired by foraging activity of birds. Each particle of PSO is influenced by the fittest particle of the swarm as well as its history while traversing in the search space. Improving the algorithm is as good as finding new algorithm, which outperforms present techniques. Improving the algorithm may further improve the quality of solution at the cost of increased complexity. Basically our motivation is real-time implementation of SA problem in hardware (which is out of scope of the present paper), where computational complexity is a major constraint. So, we are limiting our studies to original form of the algorithm. In this paper, we used Firefly algorithm to solve the spectrum allocation problem and compared its performance with basic PSO and ABC in terms of quality of solution and time complexity to obtain the results.

In recent years, Firefly algorithm has emerged as a heuristic algorithm to solve optimization problems [24]. The use of fireflies as an optimization tool was initially proposed by Yang in 2008 [23] which imitate the social behavior of fireflies, according to distinctive flashing and attraction properties of fireflies to protect themselves from predators and absorb their prey. Improved variants of FA had been proposed in order to compare the method with other meta-heuristic algorithms.

This paper aims to solve SA problem using FA and compare the performance with PSO and ABC to maximize the network capacity. The rest of the paper is described as follows. Section 2 provides the previous works related to the methods which solve spectrum allocation problem and the applications of FA. The context of FA is explained in Section 3. The SA model of cognitive radio network is described in Section 4. Cognitive radio spectrum allocation based on firefly algorithm is presented in Section 5. Section 6 provides experimental setup and simulation results followed by conclusions in Section 7.

2 Related Work

In the literature, various evolutionary algorithms were used to solve spectrum allocation problem of cognitive radio networks. Zhijin Zhao et al., solved spectrum allocation problem using GA, QGA and PSO algorithms. It has been shown that these evolutionary algorithms greatly outperform the commonly used color sensitive graph coloring algorithm [28]. Fang Ye et al., proposed an improved genetic spectrum assignment

model, in which genetic algorithm is divided into two sets of feasible spectrum assignment and randomly updates the spectrum assignment strategies [27]. The penalty function is added to the utility function to achieve spectrum assignment strategy that satisfies the interference constraints with better fitness. This method resulted better performance than the conventional genetic and quantum genetic assignment model [27]. Xiaoya Cheng et al., proved that a biological inspired ABC optimization algorithm performs better than GA for optimizing the spectrum allocation for fairness and efficiency for cognitive users [22]. This paper defines a general framework for spectrum allocation in cognitive radio system and optimize the allocation of spectrum for fairness and efficiency. An Ant Colony System (ACS) technique based on the Graph Coloring Problem (GCP) is proposed for spectrum allocation in CR network [4]. The performance of ACS was compared with PSO for various number of secondary users, primary users and available channels. ACS performed better than the other algorithms, but it requires more execution time to converge the solution. Ahmad Ghasemi et al., proposed a multi-objective spectrum allocation model is presented and a new spectrum allocation method based on Differential Evolution (DE) algorithm with the multi-objective functions as weighted summation of Max-Sum-Reward (*MSR*), Max-Min-Reward (*MMR*) and Max-Proportional-Fair (*MPF*) [6].

In this work, a new population based meta-heuristic algorithm namely FA is used, to solve the spectrum allocation problem. Nature inspired metaheuristic algorithms are becoming popular to solve global optimization problems [26]. In literature, FA is being used for solving nonlinear design, structural optimization, wireless sensor network localization and multi-modal optimization problems etc. It is reported that Firefly algorithm performs better compared to other evolutionary algorithms in terms of quality of solution and convergence rate.

FA is used to solve standard pressure vessel design problem and the solution provided by FA is far better than other algorithms, and also proposed a few test functions to validate the FA algorithm [25]. Firefly algorithm was used to solve Economic Emissions Load Dispatch Problem [3]. The experimental result has clearly shown the efficiency and success rates of the firefly algorithm compared to particle swarm optimization and genetic algorithm for solving the particular optimization problem [1]. Amir Hossein et al., applied firefly algorithm for mixed variable structural optimization [19]. The FA code was tested to solve six structural optimization problems. The optimization results indicated that FA is more efficient than other meta-heuristic algorithms such as PSO, GA, SA and HS [5]. Saibal et al., performed a comparative study of FA with PSO for noisy non-linear optimization problems [2]. It was proved that FA is more powerful in terms of time taken to solve the optimization problem because of the effect of attractiveness function which is unique to the firefly behavior [17]. The application of FA algorithm for solving spectrum allocation problem is not yet explored.

3 Firefly Algorithm

This section briefs about the nature of firefly algorithm. Fireflies produce short and rhythmic flashes. These flashes are to attract female partner (communication) and to

attract potential prey. In addition, flashing may also serve as a protective warning mechanism. Light intensity from a particular distance r from a light source obeys inverse square law, as distance increases light intensity decreases. Furthermore, the air absorbs light which becomes weaker and weaker as the distance increases. These two combined factors make most fireflies visible only to a limited distance. The FA has mainly two important issues, change in light intensity, formulation of attractiveness. The attractiveness of a firefly is calculated by its brightness, which in turn is corresponding fitness of objective function. As the light intensity and the attractiveness decreases as the distance from the source increases. So the light intensity and attractiveness are considered as monotonically decreasing functions. The light source obeys inverse square law and light intensity is a function of distance expressed as [24]

$$I(r) = I_o e^{-\gamma r^2} \quad (1)$$

where I is the light intensity, I_o is the original light intensity and γ is light absorption coefficient. As a firefly's attractiveness is proportional to the light intensity seen by adjacent fireflies, so it is defined as.

$$\beta(r) = \beta_o e^{-\gamma r^2} \quad (2)$$

where β_o is attractiveness at $r = 0$

If firefly j attracts the firefly i , then it moves towards firefly j and the state of firefly i can be described as

$$x_i = x_i + \beta_o e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \epsilon_i \quad (3)$$

where x_i and x_j are the locations of firefly i and firefly j . α is randomization parameter and ϵ_i is a vector of random numbers with uniform distribution. The pseudo code for FA is given in Algorithm 1.

4 Spectrum Allocation Model of Cognitive Radio Network

In this section we describe about the SA model of cognitive radio network architecture. Assume a network of N secondary users (1 to N), M spectrum channels and K primary users (1 to K). The general spectrum allocation model consists of channel availability matrix $L = \{l_{n,m} | l_{n,m} \in \{0, 1\}\}_{N \times M}$, where $l_{n,m} = 1$ if and only if channel m is available to user n , and $l_{n,m} = 0$ otherwise. The channel reward matrix $B = \{b_{n,m}\}_{N \times M}$ where $b_{n,m}$ represents the reward that can be obtained by user n using the channel m . The interference constraint matrix $C = \{c_{n,p,m} | c_{n,p,m} \in \{0, 1\}\}_{N \times N \times M}$ represents the interference constraint among secondary users, where $c_{n,p,m} = 1$ if users n and p would interfere with each other if they use channel m simultaneously and $c_{n,p,m} = 0$ otherwise and $c_{n,p,m} = 1 - l_{n,m}$ if $n = p$ [18].

In real-time applications, the spectrum environment changes slowly while users perform network-wide spectrum allocation operation quickly. Here we assume that the location, available spectrum, etc. are static during the spectrum allocation, thus L , B

Algorithm 1. Pseudo-code for Firefly Algorithm (FA) [24]

Step 1: Initialize the control parameter values of the FA : light absorption coefficient γ , attractiveness β , randomization parameter α , maximum number of iterations t_{max} , the number of fireflies NP , domain space D

Step 2: Define objective function $f(\vec{x})$, $\vec{x} = (x_1, x_2, x_3, \dots, x_d)$, Generate the initial location of fireflies x_i ($i = 1, 2, \dots, NP$) and set the iteration number $t = 0$

Step 3:

```

while  $t \leq t_{max}$  do
  for  $i=1$  to  $NP$  //do for each individual sequentially do
    for  $j=1$  to  $NP$  //do for each individual sequentially do
      compute light intensity  $I_i$  at  $x_i$  is determined by  $f(x_i)$ 
      if  $I_i \leq I_j$ , then
        Move firefly  $i$  towards  $j$  as described in Equation 3
      endif
      Attractiveness varies with distance  $r$  via  $e^{-\gamma r}$ 
      Evaluate new solutions and update light intensity
      Check the updated solutions are within limits
    end for
  end for
  Step 3.1:
  Rank the fireflies and find the current best;
  Increase the Iteration Count
   $t = t + 1$ 
end while

```

and C are constant in an allocation period. The conflict free spectrum assignment matrix $A = \{a_{n,m} | a_{n,m} \in \{0, 1\}\}_{N \times M}$, where $a_{n,m} = 1$ if channel m is allocated to secondary user n , and $a_{n,m} = 0$ otherwise. A must satisfy the interference constraints defined by C :

$$a_{n,m} \cdot a_{p,m} = 0, \text{ if } c_{n,p,m} = 1, \forall 1 \leq n, p \leq N, 1 \leq m \leq M \quad (4)$$

For the given L and C , spectrum allocation is to maximize network utilization $U(R)$ and the optimal conflict free channel assignment matrix A^* :

$$A^* = \underset{A \in (L,C)}{\operatorname{argmax}} U(R) \quad (5)$$

Here, we consider three fitness functions as in [28].

1. Max-Sum-Reward (MSR): It maximizes the total spectrum utilization in the system regardless of fairness. This optimization problem is expressed as:

$$MSR : U(R) = \sum_{n=1}^N \sum_{m=1}^M a_{n,m} \cdot b_{n,m} \quad (6)$$

2. Max-Min-Reward (MMR): It maximizes the spectrum utilization of the user with the least allotted spectrum. This optimization problem is expressed as:

$$MMR : U(R) = \min_{1 \leq n \leq N} \sum_{m=1}^M a_{n,m} \cdot b_{n,m} \quad (7)$$

3. Max-Proportional-Fair (MPF): It maximizes the fairness for single-hop flows and the corresponding fairness-driven utility optimization problem expressed as:

$$MPF : U(R) = \left(\prod_{n=1}^N \left(\sum_{m=1}^M a_{n,m} \cdot b_{n,m} \right) \right)^{1/N} \quad (8)$$

5 Spectrum Allocation Based on Firefly Algorithm

In the proposed FA-based spectrum allocation, each population specifies the conflict free channel assignment matrix. Here we propose to encode only those elements such that $l_{n,m} = 1$ and it refers to the dimension of the population. The value of every element in the population is randomly generated that satisfies interference constraints C . The value of L , B and C are initialized as [18]. The proposed FA-based spectrum allocation algorithm proceeds as follows:

1. Given $L = \{l_{n,m} | l_{n,m} \in \{0, 1\}\}_{N \times M}$, $B = \{b_{n,m}\}_{N \times M}$ and $C = \{c_{n,p,m} | c_{n,p,m} \in \{0, 1\}\}_{N \times N \times M}$, set the dimension of the population as $D = \sum_{n=1}^N \sum_{m=1}^M l_{n,m}$, and set $L_1 = (n, m) | l_{n,m} = 1$ such that the elements in L_1 are arranged in ascending order with n and m .
2. Initialize the control parameters of the algorithm α , β , γ and t_{max} .
3. Generate the initial location of fireflies randomly $X_i = [x_{1,i}, \dots, x_{3,i}, \dots, x_{D,i}]$ where $x_{d,i} \in 0, 1$, and $i \in (1 \dots NP)$
4. Map the population $x_{d,i}$ to $a_{n,m}$, where (n, m) is the d^{th} element of L_1 for all $d \in 1 \dots D$ and $i \in (1 \dots NP)$. The complete A matrix should satisfy the constraint matrix C , if any violations are there then one of the user will get the channel m depends on their reward value and the corresponding element of the matrix A is set to 1 or 0.
5. Compute the fitness of the each individual of the current population and ranking fireflies according to their intensity (fitness values).
6. Find the current best solution and move all fireflies to the better locations as defined in Algorithm 1.
7. If it reaches the predefined maximum generation then derive the assignment matrix as mentioned in the step 4 and stop the process else go to step 4 and continue.

6 Experimental Setup and Results

To evaluate the FA algorithm for solving spectrum allocation problem, we setup the objective functions by assuming that the network is noiseless and static environment. The entire experiment is carried out in MATLAB software. In this setup, we consider that the network has N secondary users, K primary users and M channels in network. Each primary user selects the channel from the available list and having the protection range of d_P which is constant. Each secondary user can adjust its communication range

with in the bounds of d_{min} and d_{max} to avoid the interference between the secondary and primary users.

By setting the values of $N=20$, $M=20$, $K=20$, $d_p=2$, $d_{min}=1$ and $d_{max}=5$, the channel availability, reward, constraint matrices are derived as in pseudo-code of Appendix 1 [18]. For comparison purpose the same problem was solved by using PSO and ABC algorithms. The parameters of PSO algorithm are defined as follows: Number of particles $NP=20$, maximum iterations as 500, weighting coefficients c_1 , c_2 are set to 0.9 and inertial weight $\omega=0.3$. The algorithmic parameters of ABC are set as: colony size $NP=20$, number of cycles=500 and limit=100. For FA number of fireflies $NP=20$, maximum iterations as 500, $\alpha=0.25$, $\beta=0.2$ and $\gamma=1$ [23].

Table 1. Performance analysis of FA, PSO and ABC

| Fitness Function | FA | | | PSO | | | ABC | | |
|------------------|----------|--------|-------|----------|--------|-------|----------|--------|-------|
| | Time | Reward | std % | Time | Reward | std % | Time | Reward | std % |
| <i>MSR</i> | 2.520e+4 | 2587 | 2.34 | 5.098e+4 | 2488 | 2.35 | 5.109e+4 | 2195 | 1.78 |
| <i>MMR</i> | 2.532e+4 | 60.8 | 15.18 | 5.111e+4 | 54.4 | 30.82 | 5.245e+4 | 46.5 | 8.19 |
| <i>MPF</i> | 2.525e+4 | 118.8 | 3.26 | 5.103e+4 | 101.3 | 3.89 | 5.134e+4 | 105 | 4.15 |

In this experimental setup, both algorithms were run for 20 independent runs. Individually three fitness functions *MSR*, *MMR* and *MPF* as in Equation (6), (7) and (8) are optimized using PSO, ABC and Firefly algorithms. The results are tabulated in Table.1. This table describes about the mean timing complexity of the algorithm, mean reward value (quality of service) and standard deviation (in percentage) of 20 runs. The computational timing complexities of FA, PSO and ABC are evaluated as [21]. The following test code (Algorithm 2) was used for evaluating timing complexity.

Algorithm 2. Test Code

```

for i=1 to 1000
x= double(5.55);
x=x+x; x=x./2; x=x*x; x=sqrt(x); x=log(x);
x=exp(x); y=x/x;
end for

```

The timing complexity (execution time) T of each algorithm is calculated as:

$$T = \frac{T_2 - T_1}{T_0} \quad (9)$$

where T_2 is the total execution time of the optimization problem, T_1 is the time required for objective function alone and T_0 is the execution time of test code. T_1 is obtained by evaluating each objective function for 3000 iterations and T_2 is the time for total execution time including objective function over 3000 function evaluations.

The convergence graphs under a fixed network topology is shown in Fig.1, Fig.2 and Fig.3. These figures corresponding to *MSR*, *MMR* and *MPF* objective functions respectively, optimized by FA, PSO and ABC algorithms. The FA algorithm achieved a maximum of ~17% and ~13% performance improvement in quality of service (*MPF*) and ~100%, ~103% improvement in timing complexity with respect to PSO and ABC algorithms. Fig.4 shows the convergence graphs for $N=10$, $M=10$ and $K=10$, corresponding to three fitness functions (*MSR* value is divided by number of secondary users).

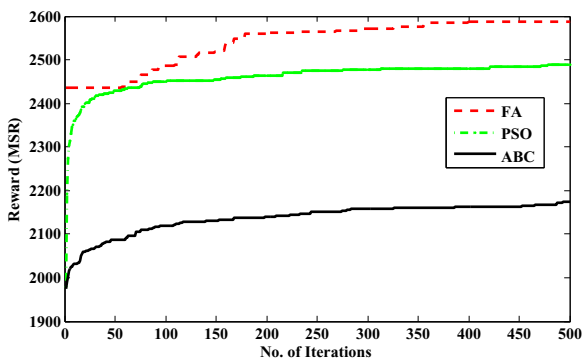


Fig. 1. Convergence Graph (Maximum Sum Reward)

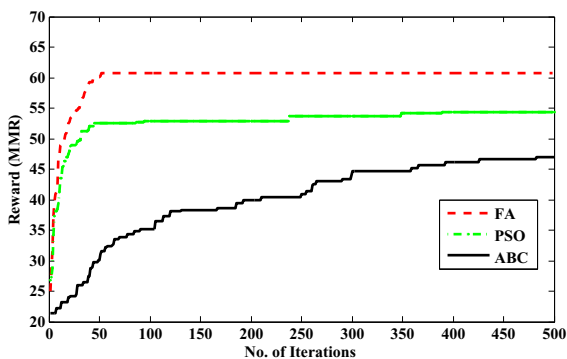


Fig. 2. Convergence Graph (Max Min Reward)

Fig.5 shows the convergence graphs for $N=5$, $M=5$ and $K=5$, corresponding to three fitness functions. In this figure, PSO performs better compared to FA and ABC, because of lower dimension of the problem. From all the convergence graphs, it can be concluded that FA has attained higher reward value. This confirms the superiority of FA algorithm having high convergence speed, quality of solution and low timing complexity with respect to PSO and ABC algorithms.

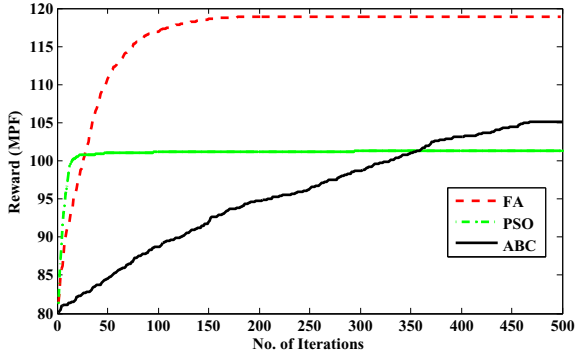


Fig. 3. Convergence Graph (Max Proportional Fair Reward)

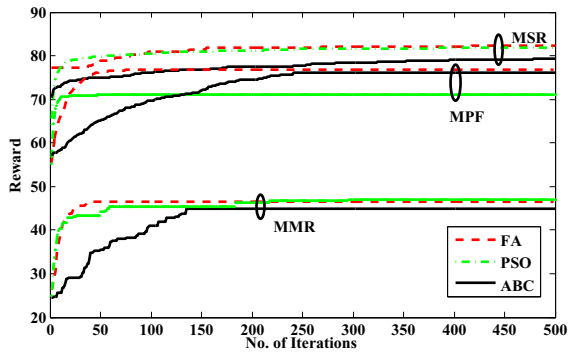


Fig. 4. Convergence Graph

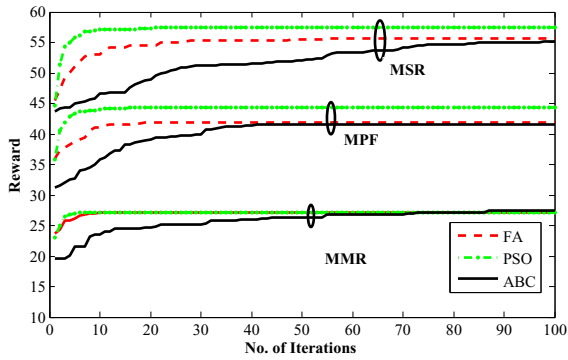


Fig. 5. Convergence Graph

7 Conclusion

Spectrum allocation (SA) for cognitive radio network is computationally complex, real time NP-hard optimization problem. In this work, the SA problem is solved by using three evolutionary methods: particle swarm optimization (PSO), Artificial Bee Colony (ABC) and firefly algorithm (FA). Recent literature reported that FA's dominance over PSO and ABC on test functions due to its unique characteristic of attraction towards global best population member. In PSO, particles are oriented towards global best particle irrespective of distance between them. But FA introduces a distance paradigm that has implicit local as well as global search motivation, thereby maintaining divergence. In this paper, we focused on performance evaluation of three methods to the SA problem in-terms of critical characteristics of optimization algorithms such as accuracy, convergence speed and repeatability. In all aspects, result shows that FA's performance is superior to that of PSO and ABC, proving maximum utilization of network capacity by optimizing *MSR*, *MMR* and *MPF* utilization objective functions and provide conflict free channel assignment to secondary users.

Acknowledgment. The authors are thankful to the University Grants Commission (UGC), Government of India for providing necessary support to carry out this research work.

References

1. Apostolopoulos, T., Vlachos, A.: Application of the firefly algorithm for solving the economic emissions load dispatch problem. *International Journal of Combinatorics* 2011 (2010)
2. Chai-Ead, N., Aungkulanon, P., Luangpaiboon, P.: Bees and Firefly algorithms for noisy non-linear optimization problems. In: *Proceedings of the International Multi Conference of Engineering and Computer Scientists*, vol. 2 (2011)
3. Dhillon, J.S., Parti, S.C., Kothari, D.P.: Stochastic economic emission load dispatch. *Electric Power Systems Research* 26(3), 179–186 (1993)
4. Koroupi, F., Talebi, S., Salehinejad, H.: Cognitive radio networks spectrum allocation: An ACS perspective. *Scientia Iranica* 19(3), 767–773 (2012)
5. Gandomi, A.H., Yang, X.-S., Alavi, A.H.: Mixed variable structural optimization using Firefly Algorithm. *Computers & Structures* 89(23), 2325–2336 (2011)
6. Ghasemi, A., Qassemi, F.: Multi-Objective Spectrum Allocation Based on Differential Evolution Algorithm between Cognitive Radio Users. *SWISS Journal of Electrical and Computer Applications* 1 (2012)
7. Haykin, S.: Cognitive Radio: Brain-Empowered Wireless Communications. *IEEE Journal on Selected Areas in Communications* 23(2), 201–220 (2005)
8. Huang, J., Berry, R.A., Honig, M.L.: Auction-Based Spectrum Sharing. *Mobile Networks and Applications* 11(3), 405–418 (2006)
9. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Techn. Rep. TR06, Erciyes Univ. Press, Erciyes (2005)
10. 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)
11. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing* 8(1), 687–697 (2008)

12. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
13. Kennedy, J.: The particle swarm: social adaptation of knowledge. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp. 303–308. IEEE (1997)
14. Cao, L., Zheng, H.: Distributed spectrum allocation via local bargaining. In: Proceedings of Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2005 (SECON), pp. 475–486 (2005)
15. McHenry, M.: Spectrum white space measurements. In: New America Foundation Broad-band Forum., vol. 1 (2003)
16. Nie, N., Comaniciu, C.: Adaptive channel allocation spectrum etiquette for cognitive radio networks. In: Proceedings of First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN), pp. 269–278 (2005)
17. Pal, S.K., Rai, C.S., Singh, A.P.: Comparative Study of Firefly Algorithm and Particle Swarm Optimization for Noisy Non-Linear Optimization Problems. *International Journal of Intelligent Systems and Applications (IJISA)* 4(10), 50 (2012)
18. Peng, C., Zheng, H., Zhao, B.Y.: Utilization and Fairness in Spectrum Assignment for Opportunistic Spectrum Access. *Mobile Networks and Applications* 11(4), 555–576 (2006)
19. Rao, S.S., Rao, S.S.: *Engineering optimization: theory and practice*. John Wiley & Sons (2009)
20. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp. 69–73. IEEE (1998)
21. 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. KanGAL Report 2005005 (2005)
22. Cheng, X., Jiang, M.: Cognitive Radio Spectrum Assignment Based on Artificial Bee Colony Algorithm. In: Proceeding of IEEE 13th International Conference on Communication Technology (ICCT), pp. 161–164 (2011)
23. Yang, X.-S.: *Nature-inspired metaheuristic algorithms*. Luniver Press (2008)
24. Yang, X.-S.: Firefly Algorithms for Multimodal Optimization. In: Watanabe, O., Zeugmann, T. (eds.) SAGA 2009. LNCS, vol. 5792, pp. 169–178. Springer, Heidelberg (2009)
25. Yang, X.-S.: Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation* 2(2), 78–84 (2010)
26. Yang, X.-S.: Review of meta-heuristics and generalised evolutionary walk algorithm. *International Journal of Bio-Inspired Computation* 3(2), 77–84 (2011)
27. Ye, F., Yang, R., Li, Y.: Genetic spectrum assignment model with constraints in cognitive radio networks. *International Journal of Computer Network and Information Security (IJCNIS)* 3(4), 39 (2011)
28. Zhao, Z., Peng, Z., Zheng, S., Shang, J.: Cognitive radio Spectrum Allocation using Evolutionary Algorithms. *IEEE Transactions on Wireless Communications* 8(9), 4421–4425 (2009)
29. Zheng, H., Peng, C.: Collaboration and fairness in opportunistic spectrum access. In: Proceedings of IEEE International Conference on Communications (ICC), vol. 5, pp. 3132–3136 (2005)

Bi-objective Optimization in Identical Parallel Machine Scheduling Problem

Sankaranarayanan Bathrinath¹, S. Saravana Sankar²,
S.G. Ponnambalam³, and B.K.V. Kannan⁴

^{1,2} Kalasalingam University, Krishanankoil, Tamil Nadu, India,

³ Department of Mechatronics, Monash University, Petaling Jaya, Malaysia,

⁴ Theni Kammavar Sangam College of Technology, Theni, Tamil Nadu, India
bathri@gmail.com

Abstract. This paper presents bi-objective identical parallel machine scheduling problem with minimization of weighted sum of makespan and number of tardy jobs simultaneously. It is a known fact that identical parallel machine scheduling problem with makespan and number of tardy jobs based criteria is NP hard. Metaheuristics has become most important choice for solving NP hard problems because of their multi-solution and strong neighborhood search capabilities in a reasonable time. In this work Simulated Annealing Algorithm (SA) and Genetic Algorithm (GA) has been proposed to optimize two different objectives namely (i) minimization of make span (ii) minimization of number of tardy jobs using combined objective function (COF). The effectiveness of the proposed algorithm have been analyzed by means of benchmark problem taken from the literatures and relative performance measures for the algorithm have also been computed and analyzed. Computational results show that GA outperforms SA by a considerable margin.

Keywords: Identical parallel machine scheduling, Genetic Algorithm, Simulated Annealing Algorithm, make span, number of tardy jobs.

1 Introduction

Scheduling deals with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives [1]. Each and every task can have different priority index level, some may have an earliest starting time and while others may have the due date criteria. The corresponding objectives can also lead to different forms and the primary objective may be the minimization of the completion time of the last job and other may be the minimization of the number of jobs completed after their corresponding due dates. In general minimizing the make span and minimizing the number of tardy jobs also minimizes the time the shop is operating which also implies to minimize the support cost as well as the maximize the use of resources. In this work we are focusing on identical parallel machine scheduling with the objective of minimizing make span and number of tardy jobs simultaneously.

Earlier Graham [2] proposed LPT, Coffman, Garey and Johnson [3] proposed MULTIFIT algorithm, Lee and Massey [4] presented the COMBINE algorithm in which the MULTIFIT has obtained the initial solution by LPT were used to minimize

make span. Later Ghomi et.al [5] and Riera et.al, [6] presented pair-wise interchange (PI) algorithm, Gupta et.al, [7] proposed LISTFIT algorithm, Lee et.al, [8] presented Simulated Annealing Approach (SAA), Ali et.al, [9] proposed discrete particle swarm optimization algorithm (DPSO), Chen et.al [10] proposed hybrid dynamic harmony search algorithm, and Liang et.al [11] proposed variable neighborhood search algorithm (VNS) for minimizing make span in single objective identical parallel machine scheduling problems.

Gupta et.al, [12] proposed lower bounds, Briand et.al, [13] presented the computation of good quality lower bound and upper bound, Baoqiang Fan et.al, introduced pseudo-polynomial-time algorithm and fully-polynomial-time approximation [14], and Yunqiang Yin et.al, proposed branch and bound procedure with two agents [15] for minimizing number of tardy jobs in the production scheduling.

In this work an attempt has been made to develop a methodology to solve two different objectives by considering them as a single objective using combined objective function (COF). The proposed algorithm is experimented with benchmark problems given in literatures and found consistent.

2 Problem Description

A set $N = \{J_1, J_2, \dots, J_n\}$ of n jobs are to be scheduled on a set $M = \{M_1, M_2, \dots, M_m\}$ of m identical parallel machines. The objective is to find the optimal schedule $S = \{S_1, S_2, \dots, S_m\}$ where S_j is a subset of jobs assigned to machine M_j such that $\max\{C_1(S), C_2(S), \dots, C_m(S)\} = C_{max}(S)$ is minimum where,

$$C_j(S) = \sum_{p_i \in S_j} p_i \quad (1)$$

While the second criterion considered in this problem is to minimize the number of tardy jobs where it is associated with each job j and its due date has to be $d_j > 0$. Let $U_j = 1$ if due date for job j is smaller than the completion time C_j of job j , otherwise $U_j = 0$. The total number of tardy jobs (N_t) is defined as

$$N_t = \sum_{j=1}^n U_j \quad (2)$$

Combined objective fitness function (COF) is obtained by combining all the above two objectives into single scalar function so as to minimize makespan and number of tardy jobs simultaneously which has been framed as:

$$\text{minimizing } (\delta * C_{max}) + (1 - \delta) * \sum_{j=1}^n U_j \quad (3)$$

Where, δ is the weight value for makespan in the objective function and $1 < \delta < 0$. Weight values have been considered to give relative importance to individual criteria in combined objective fitness function where $\delta \in \{0.3\}$. The initial sequence of jobs is allocated to the machines by means of FAM (First Available Machine) rule. The FAM rule is used to assign the unscheduled jobs to the available machine at earliest time among the all others.

The following assumptions are considered in this problem:

- i. Each job $J_i \in N$ has only one operation and a deterministic processing time (p_i) for the operation which includes any setup time required.
- ii. All the jobs are ready at time zero.
- iii. Pre-emption, division, cancellation of operations of any job is not allowed.
- iv. Each machine can process one job at a time and there is no precedence relation between jobs.

3 Proposed Methodology

In this work, Simulated Annealing Algorithm (SA) and Genetic Algorithm (GA) have been implemented to obtain optimal solution using combined objective fitness function (COF).

3.1 Proposed Simulated Annealing Algorithm (SA)

In a SA approach feasible solutions to the problem are found out using following steps.

Step 1: Initialization the initial temperature (T), annealing rate, cooling function.

Step 2: The algorithm begins with generation of initial job sequence by means of Longest Processing Time (LPT) and assign the job sequence to {S} and BEST. Obtain the combined-objective fitness function using the equation (3) for {S} by applying the First Available Machine (FAM) rule to assign the jobs over machines.

Step 3: The annealing function will then modify this schedule and return a new schedule that has been changed by an amount proportional to the temperature.

Step 4: The algorithm determines whether the new schedule {S'} is better or worse than the current {S}. If {S'} is better than the {S}, it becomes the next schedule {S=S'}. If the {S'} is worse than the {S}, the algorithm may still make it the next schedule based on an acceptance probability.

Step 5: The algorithm lowers the temperature proportional to cooling rate and store the best point found so far.

Step 6: The algorithm stops when the temperature reaches 0.

3.1.1 Parameters for Simulated Annealing Algorithm.

- (i) Initial Temperature (T_i) $\in \{1000\}$
- (ii) Final Temperature (T_f) $\in \{0\}$
- (iii) Annealing Rate (α) $\in \{0.85\}$
- (iv) Number of perturbation at each temperature drop (N_p) $\in \{10\}$

4 Proposed Genetic Algorithm (GA)

In this paper GA has been implemented to obtain optimal solution for the problem. The logical flow of the algorithm can be illustrated by the pseudo code shown below.

Step 1: Choose a Coding Scheme to represent decision variable, Choose appropriate Reproduction, Crossover and Mutation operators.

Choose population size, crossover probability, Mutation probability and the Termination criterion.

Step 2: Generate the Initial Population and evaluate the fitness values

Do while Termination Criterion is not met

Step 3: Perform Reproduction to create intermediate mating pool

Step 4: Perform Crossover to create off springs

Step 5: Perform Mutation on every string of the intermediate population

Step 6: Evaluate the strings in the new population

End do

Decode the best string in the final population to get the solution.

3.2.1 Solution Representation

The each chromosome in GA represents the job order sequence based on which jobs are assigned to the machines. For solving identical parallel machine scheduling problem using GA, the chromosomes have to be coded in the form of string structure. In this paper, phenostyle codification has been used to represent the solution for the problem. Chromosome representations have been represented using an example. If the floor has three machines and six jobs that is $N = \{J_1, J_2, \dots, J_6\}$, $M = \{M_1, M_2, M_3\}$, $\{p_i\} = \{39, 21, 22, 30, 36, 40\}$, a feasible schedule $S = \{S_1, S_2, S_3\} = \{ (J_6, J_4), (J_1, J_2), (J_3, J_5) \}$, then the solution can be represented as given in Fig.1.

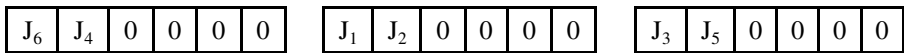


Fig. 1. Chromosome Representation

3.2.2 Fitness Evaluation

Basic nature of GA is suited for solving maximization problems. Maximization problems are usually transformed into minimization problem by suitable transformation. Fitness function $F(i)$ has been initially derived from the objective function and used in successive genetic operations. If the problem stated are maximization problem then $F(i) = O(i)$, where $O(i)$ is the objective function value of i^{th} individual and $F(i)$ is the fitness function value of i^{th} individual. In case of minimization problem then it has to be converted into equivalent maximization problem, where its fitness values are

$$F(i) = \frac{1}{f(i)} \tag{4}$$

In this paper, minimization work has been carried out, fitness value $f(i)$ has been calculated using equation 3 and 4.

3.2.3 Selection Operator

The selection operator has been modelled on the basis of Roulette wheel [16] mechanism. The probability of selection operator for each chromosome is based on a fitness value relative to the total fitness value of the population. The Fitness value for this work is $F(i) = 1/COF$, where COF is given in equation 3.

3.2.4 Crossover

Two parent strings have been selected randomly with the crossover probability and crossover site is also selected in a random manner. These two parent strings have been exchanged after the crossover site. New two child strings are developed. An example of crossover is shown in Fig. 2 and Fig. 3

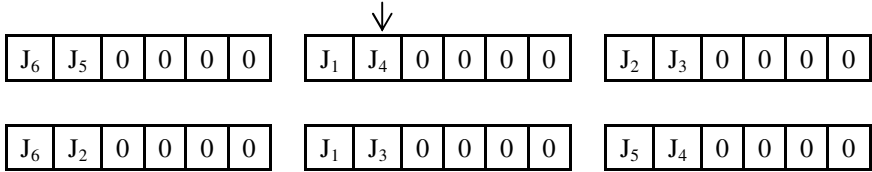


Fig. 2. Chromosomes Before Crossover

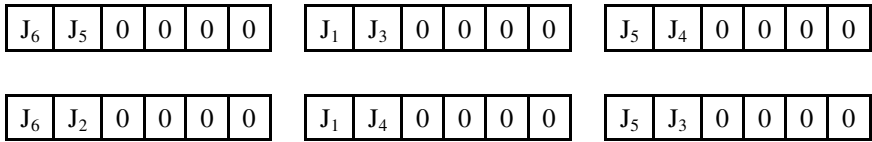


Fig. 3. Chromosomes After Crossover

3.2.5 Mutation

In this case with a very small probability, a random number ‘r’ is generated for each individual. The particular individual undergoes mutation when the mutation probability $p_m < r$. Fig. 4 and Fig. 5 shows before and after mutation process respectively.

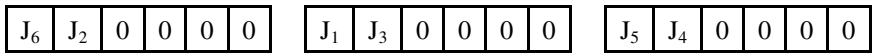


Fig. 4. Chromosomes Before Mutation

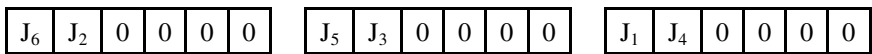


Fig. 5. Chromosomes After Mutation

3.2.6 Control Parameters

Based on the trail experiments the control parameter values for GA have been determined which produces satisfactory outputs are given as below. Population size (N_p) = 20, Cross over probability (p_c) = 0.85, Mutation probability (p_m) = (1/n) where n is the number of jobs, and Termination = stop after 100 generations.

5 Computational Experiments

Gupta et.al, [7] proposed several uniform distribution generating schemes for generating four problem sets to conduct experiments is as shown in Table. 1. Several computational experiments have been carried out using m, n and minimum and maximum values of uniform distribution used to generate processing times. For generating the due dates, it has to be lie between the processing time as well as the average value of the processing time with the tightness. From the literature [17], the tightness is found to be 1.10 which means the due date is at most as bigger as the 10% of the average processing time. The due dates are calculated using the relation taken from the literature as given by,

$$d_j = \text{random}(p_j, (\sum_{j=1}^n p_j / m) * TG) \tag{5}$$

Where, TG is the tightness.

Table 1. Summary of Computational Experiments

| Experimental names | m | n | p |
|--------------------|--------------|-----------------------------------|--------------------------------|
| E1 | 3,4,5 | 2m,3m,5m | U(1,20),U(20,50) |
| E2 | 2,3,4,6,8,10 | 10,30,50,100 | U(100,800) |
| E3 | 3,5,8,10 | 3m+1,3m+2,4m+1, 4m+2,5m+1,5m+2 | U(1,100),U(100,200),U(100,800) |
| E4 | 2 | 9 | U(1,20),U(20,50),U(50,100), |
| | 3 | 10 | U(100,200),U(100,800) |

All algorithms are coded in MATLAB 6.5 and are executed in Intel® Core™ i5 CPU M430 @ 2.27 GHz 2.27GHz with 4GB RAM.

6 Results and Discussion

The relative performance of the one Algorithm with respect to another algorithm is calculated for the proposed metaheuristics. For example, a value of c/d in GA/SA means that among the one hundred problems, there are c number of problem instances for which GA yields a better solution than SA, d problems for which SA performs better than GA and 100-c-d problems for which both GA and SA yield the same solution. The Table 2 to Table 7 shows the relative performance of the metaheuristics for the experiment levels E1, E2, E3 and E4 and its weight value has been taken as 0.3. These tables also show the computation time in seconds for the proposed SA and GA. From these tables column 12 shows the relative performance measures of GA over SA.

Table 2. Results for Experiment E1 when $\delta=0.3$

| m | n | P _{gen} | SA | | | | GA | | | | $\delta=0.3$ GA/SA |
|---|----|------------------|------|-------|------|----------|------|-------|------|----------|-----------------------|
| | | | MIN | MEAN | MAX | CPU time | MIN | MEAN | MAX | CPU time | |
| 3 | 6 | | 5.1 | 8.71 | 11.7 | 0.15 | 3.7 | 7.92 | 11.7 | 112.22 | 80/5 |
| | 9 | U(1,20) | 8 | 13.06 | 18.8 | 0.17 | 7.1 | 11.62 | 16.2 | 132.7 | 95/4 |
| | 15 | | 17.7 | 21.98 | 26.7 | 0.25 | 13.5 | 19.62 | 23.9 | 178.78 | 100/0 |
| | 6 | | 18.4 | 24.21 | 30.3 | 0.16 | 17.6 | 23.58 | 28.7 | 125.31 | 60/10 |
| | 9 | U(20,50) | 29.4 | 35.96 | 42.4 | 0.18 | 27.9 | 34.9 | 41.4 | 152.06 | 77/21 |
| | 15 | | 49.3 | 59.17 | 66.1 | 0.22 | 45.2 | 58.2 | 65.1 | 178.62 | 74/26 |
| 4 | 8 | | 6.2 | 9.55 | 13.9 | 0.19 | 4.9 | 8.64 | 12.4 | 153.88 | 80/11 |
| | 12 | U(1,20) | 9 | 14.29 | 19.7 | 0.17 | 8.3 | 12.5 | 17.2 | 169.41 | 98/2 |
| | 20 | | 18.1 | 23.96 | 28.8 | 0.3 | 15.2 | 21.06 | 26.5 | 211.97 | 99/1 |
| | 8 | | 19.4 | 24.7 | 29.5 | 0.19 | 19.6 | 24.28 | 28.8 | 145.63 | 57/31 |
| | 12 | U(20,50) | 30.3 | 36.6 | 44.5 | 0.21 | 30 | 36.17 | 42.1 | 181.58 | 52/44 |
| | 20 | | 49.2 | 59.91 | 68.8 | 0.25 | 50.6 | 59.3 | 67.3 | 199.1 | 64/34 |
| 5 | 10 | | 5.4 | 10.18 | 13.9 | 0.21 | 4.6 | 9.24 | 12.7 | 164.71 | 81/11 |
| | 15 | U(1,20) | 10.6 | 15.34 | 20.6 | 0.24 | 8.8 | 13.64 | 18 | 193.13 | 95/5 |
| | 25 | | 19.3 | 26.03 | 31.4 | 0.36 | 15.1 | 22.62 | 29.6 | 288.1 | 99/1 |
| | 10 | | 21.3 | 25.92 | 31.7 | 0.2 | 20.9 | 25.82 | 30.2 | 176.54 | 46/47 |
| | 15 | U(20,50) | 31 | 37.85 | 45.4 | 0.25 | 31.1 | 37.68 | 44.2 | 165.4 | 52/47 |
| | 25 | | 53 | 62.01 | 70.2 | 0.36 | 53.7 | 61.34 | 70.2 | 267.93 | 67/31 |

Table 3. Results for Experiment E2 when $\delta=0.3$

| m | n | P _{gen} | SA | | | | GA | | | | $\delta=0.3$ GA/SA |
|----|-----|------------------|--------|--------|--------|----------|--------|--------|--------|----------|-----------------------|
| | | | MIN | MEAN | MAX | CPU time | MIN | MEAN | MAX | CPU time | |
| 2 | 10 | | 466.5 | 682.55 | 954.2 | 0.47 | 462.7 | 669.74 | 918.8 | 125.93 | 85/15 |
| 3 | | | 321.3 | 468.59 | 620.4 | 0.18 | 311.1 | 458.26 | 614.7 | 136.99 | 73/27 |
| 2 | | | 1669.5 | 2078 | 2415.5 | 0.24 | 1663.6 | 2075.5 | 2430.4 | 212 | 70/30 |
| 3 | 30 | | 1115.3 | 1397.3 | 1622 | 0.25 | 1114.2 | 1394 | 1628.6 | 246.01 | 61/38 |
| 4 | | | 849.4 | 1063 | 1255.6 | 0.27 | 903.6 | 1062.2 | 1231.6 | 274.56 | 60/40 |
| 6 | | | 594.9 | 729.43 | 848.7 | 0.31 | 580.8 | 712.96 | 888.9 | 330.84 | 89/11 |
| 8 | | | 444.8 | 569.19 | 659.6 | 0.35 | 438.5 | 552.59 | 727.8 | 385.35 | 89/11 |
| 10 | | | 374.4 | 473.12 | 559.9 | 0.36 | 364 | 443.47 | 525.2 | 358.74 | 95/5 |
| 2 | | | 2856 | 3368.1 | 3920.3 | 0.31 | 2856.7 | 3365.6 | 3929.5 | 303.77 | 80/20 |
| 3 | 50 | U(100,800) | 1916.4 | 2263.2 | 2633 | 0.34 | 1909.7 | 2261.2 | 2655 | 388.31 | 57/43 |
| 4 | | | 1467.9 | 1712.3 | 1988 | 0.36 | 1481.9 | 1710.1 | 1986.3 | 391.43 | 55/45 |
| 6 | | | 993.3 | 1169 | 1363 | 0.43 | 981.9 | 1157.3 | 1335.4 | 472.25 | 81/19 |
| 8 | | | 761.8 | 899.2 | 1045.5 | 0.5 | 746.9 | 879.94 | 1020.4 | 540.05 | 93/7 |
| 10 | | | 623 | 737.78 | 870.4 | 0.58 | 588.9 | 703.89 | 878.5 | 614.7 | 96/4 |
| 2 | | | 6056.3 | 6789.6 | 7624 | 0.52 | 6049 | 6783.4 | 7623.1 | 635.29 | 78/22 |
| 3 | 100 | | 4070.8 | 4555 | 5124 | 0.56 | 4050.4 | 4552.5 | 5104.1 | 563.12 | 64/36 |
| 4 | | | 3064.5 | 3435.9 | 3845.2 | 0.62 | 3041.5 | 3416.1 | 3835.2 | 623.92 | 83/16 |
| 6 | | | 2065.8 | 2324.5 | 2600.9 | 0.74 | 2059.3 | 2302.6 | 2581.7 | 798.57 | 89/11 |
| 8 | | | 1563.6 | 1767.6 | 1989.8 | 0.86 | 1560.7 | 1744.6 | 1954.6 | 807.15 | 94/6 |
| 10 | | | 1279 | 1443.6 | 1622.4 | 1.01 | 1249 | 1396.9 | 1565.3 | 1082.09 | 98/2 |

Table 4. Results for Experiment E3 U(1,100) when $\delta=0.3$

| m | n | P _{gen} | SA | | | | GA | | | | $\delta=0.3$ |
|----|----|------------------|------|--------|-------|----------|------|--------|-------|----------|--------------|
| | | | MIN | MEAN | MAX | CPU time | MIN | MEAN | MAX | CPU time | GA/SA |
| | 10 | | 27.8 | 58.03 | 85.3 | 0.21 | 27.3 | 55.98 | 76.9 | 136.8 | 86/11 |
| | 11 | | 39.6 | 61.97 | 84.6 | 0.17 | 38.6 | 59.19 | 78.1 | 138.75 | 84/14 |
| 3 | 13 | | 40.3 | 75.77 | 98.8 | 0.19 | 40.6 | 75.39 | 96.1 | 152.62 | 50/43 |
| | 14 | | 51.3 | 80.08 | 116.1 | 0.19 | 49.5 | 77.38 | 113.5 | 153.32 | 89/10 |
| | 16 | | 63.4 | 89.45 | 117.5 | 0.2 | 59.5 | 87.05 | 108.3 | 164.74 | 82/16 |
| | 17 | | 57.9 | 94.08 | 119.2 | 0.22 | 55.5 | 91.36 | 117.6 | 168.58 | 83/17 |
| | 16 | | 40.6 | 58.19 | 77.2 | 0.22 | 38.9 | 56.49 | 71.4 | 189.2 | 61/36 |
| 5 | 17 | | 37.8 | 60.61 | 76.8 | 0.23 | 35.5 | 58.75 | 75.8 | 198.74 | 66/31 |
| | 21 | | 52.9 | 75.26 | 109.5 | 0.27 | 51.5 | 73.05 | 108.2 | 223.52 | 69/30 |
| | 22 | | 59.6 | 77.75 | 96.9 | 0.27 | 58.1 | 76.08 | 95.1 | 232.37 | 60/35 |
| | 26 | | 70.9 | 92.8 | 115.1 | 0.31 | 66.6 | 90.5 | 109.2 | 256.84 | 75/24 |
| | 27 | | 74.9 | 95.13 | 129.5 | 0.31 | 73.1 | 93.75 | 115.8 | 265.74 | 69/29 |
| | 25 | U(1,100) | 42.4 | 60.24 | 83.1 | 0.33 | 45.3 | 59.9 | 77.9 | 312.94 | 41/57 |
| | 26 | | 47 | 62.81 | 78.1 | 0.31 | 47 | 62.35 | 75.4 | 326.1 | 51/48 |
| 8 | 33 | | 59.4 | 78.36 | 94.1 | 0.4 | 58.9 | 77.61 | 92.4 | 387.56 | 56/43 |
| | 34 | | 61.4 | 80.3 | 101.2 | 0.38 | 59.8 | 78.54 | 92.9 | 394.99 | 64/36 |
| | 41 | | 74.9 | 96.4 | 116.8 | 0.47 | 71.5 | 94.43 | 114.4 | 464.81 | 74/24 |
| | 42 | | 79.9 | 99.82 | 118.8 | 0.46 | 78 | 98.24 | 118 | 465.83 | 65/34 |
| 10 | 31 | | 46.3 | 61.97 | 75.8 | 0.39 | 48 | 61.55 | 74.1 | 419.91 | 47/52 |
| | 32 | | 47.2 | 64.14 | 89.8 | 0.37 | 49.1 | 63.54 | 82.6 | 438.66 | 44/56 |
| | 41 | | 62.4 | 81.32 | 104.4 | 0.51 | 60 | 80.16 | 99.1 | 523.61 | 62/38 |
| | 42 | | 67.7 | 83.84 | 98.2 | 0.49 | 69.4 | 82.89 | 100.9 | 550.93 | 52/47 |
| | 51 | | 81.5 | 99.56 | 118.4 | 0.59 | 78.7 | 97.28 | 109.1 | 442.51 | 67/32 |
| | 52 | | 83.1 | 102.58 | 123.2 | 0.62 | 85.2 | 100.52 | 122.2 | 558.56 | 68/31 |

Table 5. Results for Experiment E3 U(100,200) when $\delta=0.3$

| m | n | P _{gen} | SA | | | | GA | | | | $\delta=0.3$ |
|----|----|------------------|-------|--------|-------|----------|-------|--------|-------|----------|--------------|
| | | | MIN | MEAN | MAX | CPU time | MIN | MEAN | MAX | CPU time | GA/SA |
| 3 | 10 | | 150.2 | 171.37 | 192.8 | 0.20 | 138.4 | 157.85 | 187 | 158.57 | 99/1 |
| | 11 | | 155.7 | 181.33 | 208 | 0.19 | 151.2 | 175.38 | 206 | 161.74 | 98/2 |
| | 13 | | 190.4 | 217.63 | 241.6 | 0.18 | 176.2 | 202.75 | 224.2 | 156.96 | 100/0 |
| | 14 | | 195.2 | 226.98 | 248.5 | 0.22 | 186.6 | 220.5 | 243.5 | 156.32 | 98/2 |
| | 16 | | 234.2 | 265.77 | 297 | 0.22 | 220.2 | 250.97 | 278.7 | 180.78 | 100/0 |
| | 17 | | 239.5 | 270.59 | 305.8 | 0.23 | 233.2 | 263.37 | 295.4 | 196.92 | 96/3 |
| | 16 | | 149.2 | 171.89 | 186.9 | 0.21 | 143.4 | 158.5 | 173.4 | 146.97 | 100/0 |
| 5 | 17 | | 157.7 | 176.66 | 203.6 | 0.22 | 147.7 | 167.2 | 190.1 | 191.77 | 99/1 |
| | 21 | | 196.5 | 217.9 | 235.3 | 0.27 | 184.2 | 203.09 | 226 | 191.07 | 100/0 |
| | 22 | | 199.8 | 222.86 | 247.8 | 0.27 | 184.3 | 213.07 | 244.9 | 177.76 | 100/0 |
| | 26 | | 231.2 | 264.56 | 291.1 | 0.31 | 211.6 | 250.2 | 276.9 | 154.10 | 100/0 |
| | 27 | | 241.5 | 269.75 | 291.8 | 0.31 | 232.6 | 259.81 | 285.6 | 167.82 | 99/1 |
| 8 | 25 | U(100,200) | 145.3 | 171.28 | 183.6 | 0.32 | 141.1 | 158.21 | 170.3 | 211.16 | 100/0 |
| | 26 | | 160.5 | 177.28 | 194.4 | 0.30 | 152.4 | 166.97 | 179.8 | 193.02 | 100/0 |
| | 33 | | 196.2 | 220.3 | 239.1 | 0.36 | 185.8 | 207.87 | 227.8 | 243.83 | 100/0 |
| | 34 | | 208.1 | 223.85 | 245.1 | 0.41 | 196.6 | 213.59 | 231.7 | 244.65 | 99/1 |
| | 41 | | 247.7 | 268.36 | 287.3 | 0.47 | 238.5 | 256.05 | 274.4 | 262.58 | 97/3 |
| | 42 | | 252.6 | 271.73 | 294 | 0.43 | 240.8 | 260.68 | 286.4 | 234.44 | 100/0 |
| | 31 | | 159.8 | 173.62 | 190.1 | 0.42 | 146.7 | 161.35 | 175.7 | 205.44 | 98/2 |
| 10 | 32 | | 166.5 | 179 | 210.4 | 0.43 | 153.7 | 167.79 | 178.6 | 225.48 | 100/0 |
| | 41 | | 208.2 | 223.21 | 236.5 | 0.51 | 197.6 | 210.68 | 224.8 | 249.52 | 97/2 |
| | 42 | | 211 | 226.33 | 246.3 | 0.51 | 200.3 | 215.06 | 234 | 249.81 | 99/1 |
| | 51 | | 252.1 | 271.11 | 292.2 | 0.62 | 239.9 | 258.7 | 282.9 | 303.49 | 99/1 |
| | 52 | | 261.4 | 274.27 | 288 | 0.56 | 249.2 | 263.19 | 286.3 | 323.51 | 100/0 |

Table 6. Results for Experiment E3 U(100,800) when $\delta=0.3$

| m | n | P _{gen} | SA | | | | GA | | | | $\delta=0.3$ |
|----|----|------------------|-------|--------|--------|----------|-------|--------|-------|----------|--------------|
| | | | MIN | MEAN | MAX | CPU time | MIN | MEAN | MAX | CPU time | GA/SA |
| 3 | 10 | | 319.9 | 469.68 | 666 | 0.17 | 310.1 | 459.11 | 613.3 | 87.20 | 70/29 |
| | 11 | | 365 | 533.22 | 696.8 | 0.16 | 365.1 | 521.31 | 660.4 | 146.31 | 81/19 |
| | 13 | | 426.7 | 621.44 | 884 | 0.18 | 413.7 | 611.22 | 828.5 | 160.32 | 75/25 |
| | 14 | | 398.3 | 659.59 | 839.7 | 0.19 | 400.4 | 654.2 | 807.8 | 156.90 | 67/33 |
| | 16 | | 532.3 | 733.59 | 905.8 | 0.20 | 530.6 | 726.01 | 897.1 | 172.61 | 69/30 |
| | 17 | | 578.7 | 790.34 | 1009.5 | 0.21 | 577.1 | 783.72 | 993.1 | 178.67 | 69/31 |
| | 16 | | 302.4 | 461.02 | 610.2 | 0.20 | 292.6 | 456.05 | 581.6 | 113.24 | 67/33 |
| 5 | 17 | | 376 | 498.92 | 679.9 | 0.22 | 364.4 | 496.31 | 674.8 | 194.98 | 65/35 |
| | 21 | | 412.2 | 593.01 | 732.3 | 0.26 | 408.8 | 590.67 | 749.9 | 124.07 | 59/41 |
| | 22 | | 448.4 | 630.37 | 783.4 | 0.25 | 445.2 | 625.92 | 775.4 | 153.35 | 67/33 |
| | 26 | | 598 | 742.86 | 923.4 | 0.28 | 584.6 | 742.09 | 895.9 | 226.58 | 43/56 |
| | 27 | | 621.4 | 761.12 | 899.4 | 0.25 | 617.9 | 755.43 | 968.3 | 173.07 | 67/33 |
| | 25 | U(100,800) | 392.2 | 473.37 | 589.1 | 0.29 | 364.1 | 453.71 | 582.5 | 160.22 | 92/8 |
| | 26 | | 391.3 | 487.38 | 582.7 | 0.30 | 387.5 | 473.79 | 628.1 | 216.26 | 79/21 |
| 8 | 33 | | 490.7 | 607.16 | 707.6 | 0.34 | 487.9 | 586.88 | 712.7 | 188.37 | 90/10 |
| | 34 | | 529 | 640.2 | 776.8 | 0.32 | 507.4 | 625.48 | 779.6 | 192.65 | 86/14 |
| | 41 | | 596.5 | 746.07 | 866.3 | 0.47 | 592.8 | 726.63 | 871.9 | 288.21 | 87/13 |
| | 42 | | 655.1 | 772.94 | 929.8 | 0.41 | 656.3 | 759.16 | 936.4 | 229.66 | 85/15 |
| | 31 | | 398 | 477.35 | 552.4 | 0.36 | 370.4 | 452.18 | 583.5 | 442.80 | 93/7 |
| 10 | 32 | | 398.5 | 491.42 | 567.4 | 0.37 | 382.1 | 467.1 | 581.2 | 248.74 | 90/10 |
| | 41 | | 491.9 | 614.74 | 723.6 | 0.43 | 479.3 | 586.54 | 679 | 378.55 | 97/3 |
| | 42 | | 544 | 635.85 | 741.2 | 0.44 | 531.9 | 616.09 | 809 | 452.19 | 89/11 |
| | 51 | | 647.5 | 751.58 | 847.1 | 0.45 | 602.5 | 723.93 | 851.4 | 356.15 | 95/5 |
| | 52 | | 661.5 | 771.8 | 874.8 | 0.51 | 642.8 | 746.79 | 873.9 | 284.66 | 90/10 |

Table 7. Results for Experiment E4 when $\delta=0.3$

| m | n | P _{gen} | SA | | | | GA | | | | $\delta=0.3$ |
|---|----|------------------|-------|--------|-------|----------|-------|--------|-------|----------|--------------|
| | | | MIN | MEAN | MAX | CPU time | MIN | MEAN | MAX | CPU time | GA/SA |
| 2 | 9 | U(1,20) | 12.4 | 17.73 | 24 | 0.25 | 9.7 | 15.84 | 22.2 | 154.56 | 98/1 |
| | | U(20,50) | 44.7 | 53.23 | 63.5 | 0.18 | 41.7 | 50.23 | 60.5 | 154.61 | 99/1 |
| | | U(1,100) | 46 | 73.49 | 103.1 | 0.17 | 44.7 | 71.04 | 98.6 | 147.17 | 97/1 |
| | | U(50,100) | 93.2 | 110.00 | 131.5 | 0.16 | 85.5 | 104.14 | 127.9 | 153.01 | 98/2 |
| | | U(100,200) | 175.6 | 219.20 | 251 | 0.17 | 173.3 | 207.46 | 244.9 | 136.66 | 100/0 |
| | | U(100,800) | 439.4 | 626.89 | 839.9 | 0.16 | 423.5 | 612.33 | 806.9 | 152.22 | 89/10 |
| 3 | 10 | U(1,20) | 10.6 | 14.92 | 20.2 | 0.19 | 9.5 | 13.28 | 17.9 | 123.44 | 100/0 |
| | | U(20,50) | 35.2 | 42.65 | 53.2 | 0.17 | 32.2 | 39.33 | 50.8 | 162.66 | 100/0 |
| | | U(1,100) | 27.8 | 56.52 | 85 | 0.16 | 27 | 53.81 | 81.7 | 158.80 | 87/11 |
| | | U(50,100) | 71.4 | 86.74 | 99 | 0.16 | 66.7 | 80.58 | 92.3 | 160.88 | 98/2 |
| | | U(100,200) | 150.5 | 170.54 | 198.2 | 0.18 | 135.8 | 158.75 | 194.4 | 157.98 | 99/1 |
| | | U(100,800) | 282.5 | 460.07 | 644.3 | 0.17 | 281 | 450.22 | 616.1 | 113.08 | 76/22 |

From the above results it is clear that GA is superior to SA for the identical parallel machine scheduling problem with bi-objective optimization of minimizing makespan and minimizing number of tardy jobs simultaneously.

7 Conclusion and Scope for Future Research

The problem presented in this paper belongs to NP hard and different metaheuristics have been analyzed. The effectiveness of the proposed metaheuristics is analyzed by the number of test problems taken from the literatures and a weight value for the bi-objective fitness function has been considered. The relative performance of the each metaheuristics over other has been analyzed. Some of the major findings from the present work have been stated as follows.

Combined objective fitness function proposed for scheduling the jobs in identical parallel machine scheduling problem confirms to be an effective and comprehensive measure, as multiple decisions are frequently involved in this dynamic and competitive environment. By changing the weight values different set of optimal solutions can be obtained.

For considering the bi-objective identical parallel machine scheduling problem, the metaheuristics GA and SA have been proposed and analyze the effectiveness of the proposed algorithm and found that the GA gives the better schedule with minimum performance measures considered.

The outcome of this research leaves scope for further research towards employing a local search mechanism to further optimize the optimal solution. The proposed algorithm can be extended towards non identical parallel machines. Multi objective optimization algorithm can also be developed to produce a pareto optimal front.

References

1. Pinedo, M.L.: Scheduling: Theory, Algorithms and Systems. Springer (2012)
2. Graham, R.L.: Bounds on Multiprocessor Timing Anomalies. *SIAM Journal of Applied Mathematics* 17, 416–429 (1969)
3. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP Completeness. Freeman, San Francisco (1979)
4. Lee, C.Y., Massey, J.D.: Multiprocessor Scheduling: Combining LPT and MULTIFIT. *Discrete Applied Mathematics* 20, 233–242 (1988)
5. Fatemi Ghomi, S.M.T., Jolai Ghazvini, F.: A pairwise interchange algorithm for parallel machine scheduling. *Production Planning and Control* 9(7), 685–689 (1998)
6. Riera, J., Alcaide, D., Sicilia, J.: Approximate algorithms for the P||Cmax problem. *TOP* 4, 345–359 (1996)
7. Gupta, J.N.D., Ruiz-Torres, A.J.: A LISTFIT heuristic for minimizing makespan on Identical Parallel Machines. *Production Planning and Control* 12, 28–36 (2001)
8. Lee, W.C., Wu, C.C., Chen, P.: A Simulated Annealing approach to make span minimization on Identical Parallel Machines. *International Journal of Advanced Manufacturing Technology* 31, 328–334 (2006)

9. Ali, H.K., Behrooz, K.: A Discrete particle Swarm Optimization algorithm for Scheduling Parallel Machines. *Computers and Industrial Engineering* 56, 216–223 (2009)
10. Chen, J., Pan, Q.-K., Wang, L., Li, J.-Q.: A hybrid dynamic harmony search algorithm for identical parallel machines scheduling. *Engineering Optimization* 44(2), 209–224 (2012)
11. Liang, Y.-C., Hsiao, Y.-M., Tien, C.-Y.: Metaheuristics for drilling operation scheduling in Taiwan PCB industries. *Int. J. Production Economics* 141(1), 189–198 (2013)
12. Gupta, J.N.D., Ruiz-Torres, A.J., Webster, S.: Minimizing maximum tardiness and number of tardy jobs on parallel machines subject to minimum flow-time. *Journal of the Operational Research Society* 54, 1263–1274 (2003)
13. Briand, C., Ourari, S.: Minimizing the number of tardy jobs for the single machine scheduling problem: MIP-based lower and upper bounds. *RAIRO - Operations Research* 47(01), 33–46 (2013)
14. Fan, B., Yuan, J., Li, S.: Bi-criteria scheduling on a single parallel-batch machine. *Applied Mathematical Modelling* 36(3), 1338–1346 (2012)
15. Yin, Y., Wu, C.-C., Wu, W.-H., Hsu, C.-J., Wu, W.-H.: A branch-and-bound procedure for a single-machine earliness scheduling problem with two agents. *Applied Soft Computing* 13(2), 1042–1054 (2013)
16. Goldberg, D.E.: *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, USA (1989)
17. Samur, S., Bulkan, S.: An Evolutionary Solution to a Multi-objective Scheduling Problem. In: *Proceedings of the World Congress on Engineering*, London, UK, June 30-July 02, vol. 3 (2010)

Teaching-Learning-Based Optimization Algorithm in Dynamic Environments

Feng Zou^{1,2}, Lei Wang^{1,*}, Xinhong Hei¹, Qiaoyong Jiang¹, and Dongdong Yang¹

¹School of Computer Science and Engineering, Xi'an University of Technology
Xi'an 710048, P.R. China

²School of Physics and Electronic Information, HuaiBei Normal University
Huaibei, 235000, P.R. China
{zfemail, wangleeei}@163.com

Abstract. In many real world problems, optimization problems are non-stationary and dynamic. Optimization of these dynamic optimization problems requires the optimization algorithms to be able to find and track the changing optimum efficiently over time. In this paper, for the first time, a multi-swarm teaching-learning-based optimization algorithm (MTLBO) is proposed for optimization in dynamic environment. In this method, all learners are divided up into several subswarms so as to track multiple peaks in the fitness landscape. Each learner is learning from the teacher and the mean of his or her corresponding subswarm instead of the teacher and the mean of the class in teaching phase, and then learners learn from interaction between themselves in their corresponding subswarm in learning phase. Moreover, all subswarms are regrouped periodically so that the information exchange is made with all the learners in the class to achieve proper exploration ability. The proposed MTLBO algorithm is evaluated on moving peaks benchmark problem in dynamic environments. The experimental results show the proper accuracy and convergence rate for the proposed approach in comparison with other well-known approaches.

Keywords: Teaching-Learning-Based Optimization Algorithm, Dynamic Environments, Multi-swarm, Moving Peaks Benchmark problem.

1 Introduction

Many real world optimization problems are dynamic in which global optimum and local optima change over time. These dynamic optimization problems (DOPs) vary when the environment changes over time. Algorithms which are designed for optimizing in dynamic environments have some principles that distinguish them from algorithms designed in static environment. Finding global optimal value is not sufficient for optimizing in dynamic environments and tracking optimum during the changes in the environment is also important.

During recent years, various evolutionary computing and swarm intelligence methods, such as Genetic Algorithm [1], Evolution Strategy [2], Genetic Programming

* Corresponding author.

[3], Particle Swarm Optimization [4], Differential Evolution [5], Immune-based Algorithm [6], An Ant Colony Optimization [7], Estimation of Distribution Algorithm [8], Cultural Algorithm [9], Firefly Algorithm [10], have been proposed for solving optimization problems in dynamic environments. They have different design philosophies and characteristics. A survey of the research done on optimization in dynamic environments over the past decade is provided in [11]. An extensive performance evaluation and comparison of 13 leading evolutionary algorithms with different characteristics is presented by using the moving peaks benchmark and by varying a set of problem parameters in [12].

It can be seen that, in order to design these algorithms for dynamic environments, two important points have to be taken into consideration: diversity loss and outdated memory. The problem of outdated memory is usually solved by either simply setting each individual's memory position to its current position, or by reevaluating every memory position and setting it to either the old memory or current individual position, whichever is better. Diversity loss happens at environment change when swarm is converging on a peak. According to diversity loss, there are various methods for generating or maintaining population's diversity in dynamic environments. Reported mechanisms in the literatures for either diversity maintenance or diversity enhancement have been classified into four groups in [13, 14]: maintenance diversity after a change, maintenance diversity throughout the run, multi-swarm approaches, memory-based approaches.

TLBO [15, 16] is a new meta-heuristic algorithm with a great potential for discovering multiple optima simultaneously due to their ability of keeping good balance between convergence and diversity maintenance. In this paper, for the first time, a multi-swarm teaching-learning-based optimization algorithm (MTLBO) is proposed for optimization in dynamic environment. The proposed MTLBO algorithm divided up the population into several subswarms so as to track multiple peaks in the fitness landscape and all subswarms are regrouped in certain interval generations (regrouping period) so that the information exchange is made with all the learners in the class to achieve proper exploration ability. The proposed approach is evaluated on the moving peaks benchmark (MPB) problem [17] in dynamic environments.

The paper is organized as follows. Section 2 briefly describes the teaching-learning-based optimization algorithm. Section 3 presents the proposed algorithm in a dynamic environment. Section 4 gives the results of the proposed algorithm on the moving peaks benchmark problem and compares it with some approaches from the literature. Finally, Section 5 draws some conclusions and gives directions for future research regarding the MTLBO.

2 Teaching-Learning-Based Optimization

Rao et al. [15, 16] first proposed a novel teaching-learning-based optimization (TLBO) which mimics teaching-learning process in a class between the teacher and the students (learners). Like other nature-inspired algorithms, TLBO is also a population based method which uses a population of solutions to proceed to the global

solution. TLBO has emerged as one of the most simple and efficient techniques for solving single-objective benchmark problems and real life application problems since it has been empirically shown to perform well on many optimization problems [18, 19, 20, 21, 22]. These are precisely the characteristics of TLBO that make it attractive to extend it to solve multi-objective optimization problems [23, 24, 25].

For TLBO, the population is considered as a group of learners or a class of learners and different design variables will be analogous to different subjects offered to learners. The learners' result is analogous to the "fitness", as in other population based optimization techniques. The teacher is considered as the best solution obtained so far. Moreover, learners also learn from interaction between themselves, which also helps in their results. The process of working of TLBO is divided into two parts: Teacher Phase and Learner Phase.

2.1 Teaching Phase

A good teacher is one who brings his or her learners up to his or her level in terms of knowledge. But in practice this is not possible and a teacher can only move the mean of a class up to some extent depending on the capability of the class. This follows a random process depending on many factors.

Let M_i be the mean and T_i be the teacher at any iteration i . T_i will try to move mean M_i towards its own level, so now the new mean will be T_i designated as M_{new} . The solution is updated according to the difference between the existing and the new mean given by

$$Difference_Mean_i = r * (M_{new} - T_F \cdot M_i) \quad (1)$$

where T_F is a teaching factor that decides the value of mean to be changed, and r is a random number in the range [0, 1]. The value of T_F can be either 1 or 2, which is again a heuristic step and decided randomly with equal probability as

$$T_F = round[1 + rand(0, 1)] \quad (2)$$

This difference modifies the existing solution according to the following expression

$$newX_i = newX_i + Difference_Mean_i \quad (3)$$

Hence, learner modification is expressed as

$$T_F = round[1 + rand(0, 1)]$$

for $p = 1 : P_n$

$$Difference_Mean_i = r_i * (M_{new} - T_F * M_i)$$

$$X_{new,p} = X_p + Difference_Mean_i$$

endfor

Accept X_{new} if it gives a better function value

2.2 Learning Phase

Learners increase their knowledge by two different means: one through input from the teacher and other through interaction between themselves. A learner interacts randomly with other learners with the help of group discussions, presentations, formal communications, etc. A learner learns something new if the other learner has more knowledge than him or her. Learner modification is expressed as

```

for  $i = 1:P_n$ 
  Randomly select one learner  $X_j$ , such that  $i \neq j$ 
  if  $f(X_i) < f(X_j)$ 
     $X_{new,i} = X_{old,i} + r_i \times (X_i - X_j)$ 
  else
     $X_{new,i} = X_{old,i} + r_i \times (X_j - X_i)$ 
  endif
endfor
Accept  $X_{new}$  if it gives a better function value

```

3 The Proposed Algorithm (MTLBO)

In this section, a multi-swarm teaching-learning-based optimization algorithm (MTLBO) is proposed and used to solve dynamic optimization problems in a dynamic environment.

3.1 Multi-Swarm

The main idea of multi-swarm algorithms [26] is to divide up the swarm into several subswarms in order to track multiple peaks in the fitness landscape. In this way, the different subswarms maintain information about several promising regions of the search space, and can be regarded as a kind of diverse, self-adaptive memory.

Even though it has been shown the better performance of TLBO method over other nature-inspired optimization methods in some areas, it is necessary to modify it for optimization in dynamic environments. In this paper, a multi-swarm teaching-learning-based optimization algorithm (MTLBO) is presented. In this method, the learners are divided up into small sized subswarms in order to increase diversity. Each subswarm uses its own members to search for better area in the search space. Once subswarms are constructed, instead of the teacher and the mean of the class, each learner is learning from the teacher and the mean of his or her corresponding subswarm in teaching phase. Then each learner is learning in his or her corresponding subswarm in learning phase.

3.2 Periodic Regrouping

Since the learners of each subswarm are learning from the mean of their own corresponding subswarm, they may easily converge to a local optimum. In fact, there will be not enough information exchange among the small sized subswarms if the subswarm structures are not changed, thus the algorithm will be performed by means of these subswarms searching in parallel.

In our algorithm, the learners are regrouped dynamically [26] in certain interval generations (regrouping period) and start learning using a new configuration of small swarms. In this way, the information obtained by each subswarm is exchanged among the small sized subswarms, simultaneously the diversity of the population is increased. In fact, the subswarm structures are regrouped less frequently (regrouping period being higher) helping the algorithm to have high exploitation and high convergence rate. Otherwise, the algorithm has more exploration and diversity to cover more search space. Hence a proper regrouping period (set 5 in our case) helps to balance the exploitation and exploration of the algorithm, thus to have better performance compared with the original TLBO.

3.3 The Framework of MTLBO for DOPs

In MTLBO, we use the teacher of all learners as the monitoring point to detect the environmental changes. Before updating the teacher, we reevaluate its fitness value. If its fitness changes, it indicates that an environmental change occurs. Once an environmental change is detected, MTLBO needs to reevaluate fitness values of all learners. Otherwise, if it fails to detect the change, the previous learners will be used in the new environment. The framework of MTLBO for DOPs is shown in Algorithm 1.

Algorithm 1: MTLBO()

```

1  Begin
2  Initialize parameters
3  Initialize learners X and evaluate all learners
X
4  Divide the learners into small sized subswarms
5  Donate the Teacher and the Mean of each sub-
warm
6  while(stopping condition not met)
7  Perform Teaching phase in each subswarm
8  Perform Learning phase in each subswarm
9  if the environmental change detected
10     reevaluate all learners
11  endif
12  if mod(gen, Period) == 0
13     Randomly Regrouping all learner and Di-
viding into subswarms
14     end
15  endwhile
16 end

```

4 Experimental Analyses

To evaluate the efficiency of the proposed algorithm, the proposed algorithm was compared with some well-known algorithms on the well-known Moving Peaks benchmark (MPB) for dynamic environments.

4.1 Moving Peaks Benchmark Problem

In recent years, there has been a growing interest in studying EAs for DOPs due to their importance in real world applications since many real world optimization problems are DOPs. Over the years, a number of dynamic problem generators have been used in the literature to create dynamic test environments to compare the performance of EAs. The MPB problem proposed by Branke [17] has been widely used as dynamic benchmark problem in the literature. Within the MPB problem, the optima can be varied by three features, i.e., the location, height, and width of peaks. For the D -dimensional landscape, the problem is defined as follows:

$$F(\vec{x}, t) = \max_{i=1,2,\dots,p} \frac{H_i(t)}{1 + W_i(t) \sum_{j=1}^D (x_j(t) - X_{ij}(t))^2} \tag{4}$$

where $H_i(t)$ and $W_i(t)$ are the height and width of peak i at time t , respectively, and $X_{ij}(t)$ is the j th element of the location of peak i at time t . The p independently specified peaks are blended together by the “max” function.

The position of each peak is shifted in a random direction by a vector \vec{v}_i of a distance s (s is also called the shift length, which determines the severity of the problem dynamics), and the move of a single peak can be described as follows:

$$\vec{v}_i(t) = \frac{s}{|\vec{r} + \vec{v}_i(t-1)|} ((1-\lambda)\vec{r} + \lambda\vec{v}_i(t-1)) \tag{5}$$

where the shift vector $\vec{v}_i(t)$ is a linear combination of a random vector \vec{r} and the previous shift vector $\vec{v}_i(t-1)$ and is normalized to the shift length s . The correlated parameter λ is set to 0, which implies that the peak movements are uncorrelated.

The multi-dimensional problem space of the moving peaks function contains several peaks of variable height, width and shape. These move around with height and width changing periodically. The default settings and definition of the benchmark used in the experiments of this paper can be found in Table 1, which are the same as in all the involved algorithms.

Standard performance measures of dynamic evolutionary algorithms have been used to evaluate the performance of the proposed algorithm. The offline error [27] is defined as the average of the current errors over the entire run, where the current error is defined as the smallest error found since the last change in the environment. The mean offline error is evaluated by Eq. (10)

$$Offline_error = \frac{1}{K} \sum_{k=1}^K (H_k - F_k) \tag{10}$$

where K is the total number of environments for a run, H_k is the optimum value of the k th environment and F_k is the best solution obtained by an algorithm just before the k th environmental change.

Table 1. Default Settings for the MPB Problem

| Parameters | Value |
|--|---------|
| p (Number of peaks) | 10 |
| D (Number of dimensions) | 5 |
| s (the range of allele value) | [0,100] |
| Change frequency | 5000 |
| H_0 (the initial height for all peaks) | 50.0 |
| W_0 (the initial width for all peaks) | 0 |
| H (the range of the height of peaks shifted) | [30,70] |
| W (the range of the width of peaks shifted) | [1,12] |
| Height severity | 7.0 |
| Width severity | 1.0 |
| shift severity | 1.0 |

4.2 Experimental Settings

For all experiments, unless stated otherwise, the parameters have been set as follows: the search space has $D=5$ dimensions within $[0, 100]$, there is $p=10$ peaks, the peak heights vary randomly in the interval $[30, 70]$, and the peak width parameters vary randomly within $[1, 12]$. The peaks change position every 5000 evaluations by a distance of $S=1$ in a random direction, and their movements are uncorrelated (the MPB coefficient $\lambda = 0$). For each run of an algorithm, there were $K = 100$ environments, which result in $K \times U = 5 \times 10^5$ fitness evaluations. For each experiment, 20 independent runs are executed, and the results are the average of 20 independent runs. The number of learners is set to be 20.

4.3 Effect of Varying the Shift Severity s

As we know, the peaks are more and more difficult to track with the increasing of the shift length. Hence the performance of all algorithms degrades when the shift length increases. In this group of experiments, we compare the performance of MTLBO with SPSO, mQSO and CPSO on the MPB problems with different problem settings in terms of the shift severity, in order to detect the robustness of tracking and locating multiple optima in changing environments. The experimental results in terms of the offline error and standard deviation are shown in Table 2. The experimental results of SPSO, mQSO and CPSO are taken from the corresponding paper [28].

From Table 2 and Fig.1, it can be observed that the proposed algorithm has much better performance than other algorithms. In fact, the results show the proposed

algorithm is robust with number of peaks and can achieve acceptable results when the number of peaks is low or when the number of peaks is high. We can conclude that MTLBO is more adaptable of tracking the changing optima on the MPB problems with different problem settings in terms of the shift severity.

Table 2. Offline error±Standard error on the MPB with different shift severities

| <i>s</i> | MTLBO | SPSO | mQSO | CPSO(70,3) |
|----------|-----------|-----------|-----------|------------|
| 0 | 0.10±0.01 | 0.95±0.08 | 1.18±0.07 | 0.80±0.21 |
| 1 | 0.31±0.16 | 2.51±0.09 | 1.75±0.06 | 1.056±0.24 |
| 2 | 0.42±0.12 | 3.78±0.09 | 2.40±0.06 | 1.17±0.22 |
| 3 | 0.84±0.47 | 4.96±0.12 | 3.00±0.06 | 1.36±0.28 |
| 4 | 1.06±0.35 | 2.56±0.13 | 3.59±0.10 | 1.38±0.29 |
| 5 | 1.19±0.61 | 6.76±0.15 | 4.24±0.10 | 1.58±0.32 |
| 6 | 1.52±0.33 | 7.68±0.16 | 4.79±0.10 | 1.53±0.29 |

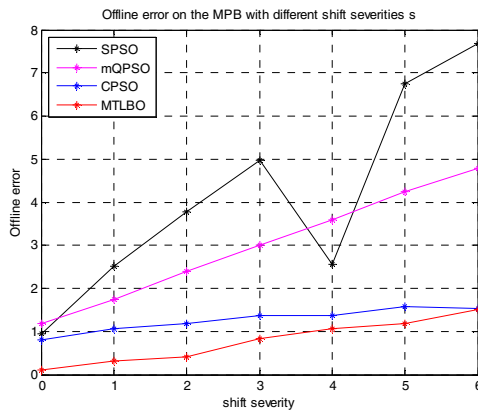


Fig. 1. Offline error on the MPB with different shift severities *s*

4.4 Effect of Varying the Peaks *p*

This set of experiments is designed to investigate the performance of MTLBO with different peaks *p* on the MPB problem. The number of peaks is set to different values in {1, 2, 5, 7, 10, 20, 30, 40, 50, 100, 200}. The experimental results in terms of the offline error and standard deviation are shown in Table 2. The experimental results of SPSO, mQSO and CPSO are taken from the corresponding paper [27]. Fig.2 presents the convergence characteristics of MTLBO for each test function.

Table 3. Offline error±Standard error on the MPB with different number of peaks

| p | MTLBO | SPSO | mQSO | CPSO |
|-----|-----------|-----------|-----------|------------|
| 1 | 0.05±0.03 | 2.64±0.10 | 5.07±0.17 | 0.14±0.11 |
| 2 | 0.08±0.04 | 2.31±0.11 | 3.47±0.23 | 0.20±0.19 |
| 5 | 0.16±0.08 | 2.15±0.07 | 1.81±0.07 | 0.72±0.30 |
| 7 | 0.08±0.03 | 1.98±0.04 | 1.77±0.07 | 0.93±0.30 |
| 10 | 0.11±0.02 | 2.51±0.09 | 1.80±0.06 | 1.056±0.24 |
| 20 | 0.13±0.05 | 3.21±0.07 | 2.42±0.07 | 1.59±0.22 |
| 30 | 0.14±0.06 | 3.64±0.07 | 2.48±0.07 | 1.58±0.17 |
| 40 | 0.15±0.04 | 3.85±0.08 | 2.55±0.07 | 1.51±0.12 |
| 50 | 0.15±0.03 | 3.86±0.08 | 2.50±0.06 | 1.54±0.12 |
| 100 | 0.14±0.06 | 4.01±0.07 | 2.36±0.04 | 1.41±0.08 |
| 200 | 0.12±0.04 | 3.82±0.05 | 2.26±0.03 | 1.24±0.06 |

From Table 3 and Fig.2, it can be seen that the performance of MTLBO is not influenced too much when the number of peaks is increased. The offline error performances of MTLBO are small but their standard error performances are too much influenced. The experiment confirms that the results achieved by MTLBO are better than the results of the other three algorithms on the MPB problems with different peaks. We can conclude that MTLBO is able to cope well with environments consisting of a large number of peaks, too.

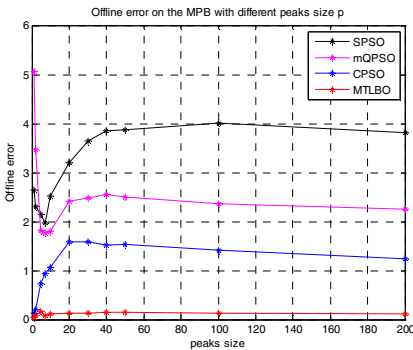


Fig. 2. Offline error on the MPB with different peaks size p

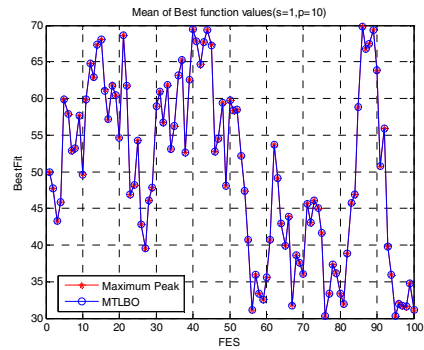


Fig. 3. Performance curves on the MPB with $s=1$ and $p=10$

5 Conclusion

In this paper, a multi-swarm teaching-learning-based optimization algorithm (MTLBO) is firstly proposed for optimization in dynamic environment. The proposed approach was evaluated and compared with several well-known methods on various configuration of the moving peak benchmark problem in dynamic environments. The experimental results show the proper accuracy and convergence rate for the proposed approach in comparison with other well-known approaches and the proposed approach is more effective in dealing with the MPB Problem.

Also, there are some relevant works to pursue in the future. Firstly, a hybrid of some diversity maintenance techniques may be used to maintain the population diversities. Secondly, neighborhood-based approaches or memetic extensions can be done to improve local search ability of algorithm in order to prevent the algorithm into a local optimum. Finally, these approaches may be used to solve CEC 2009 problems.

Acknowledgment. We are grateful to the anonymous referees for their constructive comments to improve the paper. This research was partially supported by National Natural Science Foundation of China (No. 61272283, 61100173, 61304082). This work is partially supported by the Natural Science Foundation of Anhui Province (Grants No.1308085MF82) and Doctoral Innovation Foundation of Xi'an University of Technology.

References

1. Yang, S., Tinós, R.: A Hybrid Immigrants Scheme for Genetic Algorithms in Dynamic Environments. *International Journal of Automation and Computing* 4(3), 243–254 (2007)
2. Schönemann, L.: Evolution Strategies in Dynamic Environments. *Evolutionary Computation in Dynamic and Uncertain Environments* 51, 51–77 (2007)
3. Hu, J., Li, S., Goodman, E.: Evolutionary Robust Design of Analog Filters Using Genetic Programming. *Evolutionary Computation in Dynamic and Uncertain Environments* 51, 479–496 (2007)
4. Eberhart, R.C., Shi, Y.: Tracking and optimizing dynamic systems with particle swarms. In: *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 94–100. IEEE (2001)
5. Mendes, R., Mohais, A.S.: DynDE: a Differential Evolution for Dynamic Optimization Problems. In: *The 2005 IEEE Congress on Evolutionary Computation*, vol. 3, pp. 2808–2815 (2005)
6. Trojanowski, K., Wierzchon, S.T.: Immune-based algorithms for dynamic optimization. *Information Sciences* 179(10), 1495–1515 (2009)
7. Guntsch, M., Middendorf, M., Schmeck, H.: An Ant Colony Optimization Approach to Dynamic TSP. In: *Proceedings of the Genetic and Evolutionary Conference (GECCO 2001)*, pp. 860–867 (2001)
8. Yang, S., Yao, X.: Experimental study on population-based incremental learning algorithms for dynamic optimization problems. *Soft Computing* 9(11), 815–834 (2005)

9. Saleem, S., Reynolds, R.: Cultural Algorithms in Dynamic Environments. In: Proceedings of the 2000 Congress on Evolutionary Computation, vol. 2, pp. 1513–1520 (2000)
10. Nasiri, B., Meybodi, M.R.: Speciation based firefly algorithm for optimization in dynamic environments. *International Journal of Artificial Intelligence* 8(12), 118–132 (2012)
11. Cruz, C., Gonzalez, J.R., Pelta, D.A.: Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Computing* 15(7), 1427–1448 (2011)
12. Ayvaz, D., Topcuoglu, H.R., Gurgen, F.: Performance evaluation of evolutionary heuristics in dynamic environments. *Applied Intelligence* 37(1), 130–144 (2012)
13. Branke, J.: Evolutionary approaches to dynamic optimization problems—updated survey. In: GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems, pp. 27–30 (2001)
14. Jin, Y., Branke, J.: Evolutionary Optimization in Uncertain Environments—A Survey. *IEEE Transactions on Evolutionary Computation* 9(3), 303–317 (2005)
15. 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)
16. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching-learning-based optimization: An optimization method for continuous non-linear large scale problems. *Information Sciences* 183(1), 1–15 (2012)
17. Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1875–1882. IEEE Service Center, Piscataway (1999)
18. Togan, V.: Design of planar steel frames using teaching-learning based optimization. *Engineering Structures* 34, 225–232 (2012)
19. Rao, R.V., Patel, V.: An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. *International Journal of Industrial Engineering Computations* 3, 535–560 (2012)
20. Jadhav, H.T., Chawla, D., Roy, R.: Modified Teaching Learning Based Algorithm for Economic Load Dispatch Incorporating Wind Power. In: The 11th International Conference on Environment and Electrical Engineering (EEEIC), pp. 397–402 (2012)
21. Amiri, B.: Application of Teaching-Learning-Based Optimization Algorithm on Cluster Analysis. *Journal of Basic and Applied Scientific Research* 2(11), 11795–11802 (2012)
22. Naik, A., Parvathi, K., Satapathy, S.C., Nayak, R., Panda, B.S.: QoS Multicast Routing Using Teaching Learning Based Optimization. In: Aswatha Kumar, M., Selvarani, R., Suresh Kumar, T.V. (eds.) Proceedings of ICAdC. AISC, vol. 174, pp. 49–55. Springer, Heidelberg (2013)
23. Nayak, M.R., Nayak, C.K., Rout, P.K.: Application of Multi-Objective Teaching Learning Based Optimization Algorithm to Optimal Power Flow Problem. *Procedia Technology* 6, 255–264 (2012)
24. Rao, R.V., Patel, V.: Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm. *Applied Mathematical Modelling* 37(3), 1147–1162 (2013)
25. Niknam, T., Azizipanah-Abarghooee, R., Narimani, M.R.: A new multi objective optimization approach based on TLBO for location of automatic voltage regulators in distribution systems. *Engineering Applications of Artificial Intelligence* 25(8), 1577–1588 (2012)

26. Liang, J.J., Suganthan, P.N.: Dynamic multi-swarm particle swarm optimizer. In: Proceedings 2005 IEEE Swarm Intelligence Symposium, SIS 2005, pp. 124–129 (2005)
27. 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, Department of Computer Science, University of Leicester, U.K (2008)
28. Yang, S.: A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments. *IEEE Transactions on Evolutionary Computation* 14(6), 959–974 (2010)

A Novel Ant Colony Optimization Algorithm for the Vehicle Routing Problem

Srinjoy Ganguly¹ and Swagatam Das²

¹Dept. of Electronics & Telecommunication Engineering, Jadavpur University, Kolkata, India
srinjoy_ganguly92@hotmail.com

²Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, India
swagatam.das@isical.ac.in

Abstract. The Vehicle Routing Problem (VRP) is one of the most important problems in the field of Operations Research and logistics. This paper presents a novel Ant Colony Optimization algorithm abbreviated as ACO_PLM to solve the Vehicle Routing Problem efficiently. By virtue of this algorithm we wish to propose novel pheromone deposition, local search & mutation strategies to solve the VRP efficiently and facilitate rapid convergence. The ACO_PLM provides better results compared to other heuristics, which is apparent from the experimental results and comparisons with other existing algorithms when tested on the twelve benchmark instances.

Keywords: VRP, ant system, ACO_PLM, pheromone deposition, sequential local search heuristic, intra-route mutation, inter-route mutation.

1 Introduction

The Vehicle Routing Problem [1] is a problem of representative logistics that has been studied widely in Operations Research literature over the past fifty years. The classical Vehicle Routing Problem (VRP) is defined as follows: several vehicles are deployed, each with a fixed capacity and must deliver order quantities of goods to n customers from a single depot ($i=0$). Knowing the distance between customers and , the objective of the problem is to minimize the total distance travelled by the vehicles in such a way that only one vehicle handles the deliveries for a given customer and the total quantity of goods that a single vehicle delivers do not be larger than . As the VRP is an NP-hard problem, real world instances having more than 50-75 customers cannot be solved by means of exact algorithms. However, researchers have developed several heuristics such as basic constructive mechanisms (Clarke & White savings algorithm [2], 2-opt local search [3]), simulated annealing (Chiang and Russel [4], Osman [5]), genetic algorithms (Berger et al. [6], Tan et al. [7], Osman et al. [8]), particle swarm optimization (Marinakis et al. [9], V Kachitvichyanukul [10]), ant colony optimization (Doerner et al.[11] etc which have been used many times in the past to solve instances of the VRP to near optimality. Among these heuristics, the ant colony optimization (ACO) algorithm proposed by Dorigo et al. [12], has been

successfully utilized in the past to solve a variety of NP-hard combinatorial optimization problems such as the Travelling Salesman Problem [13], the Flow-shop Scheduling Problem [14-15], the VLSI routing problem [16], etc. The ACO essentially mimics the food-gathering dynamics of the ant colony, wherein the ants communicate with each other by means of stigmergy [17].

Several versions of the VRP have been studied in literature, each of which can be encoded using the ACO by considering the depot as the nest and the customers as the food. This may provide efficient solutions as the VRP is very similar to the food-seeking behavior of the ant colonies in nature. One of the earliest ACO algorithms that could be used to solve the VRP was proposed by Bullenheimer et al. [18]. They proposed a hybrid ant system algorithm with the 2-opt and the saving algorithm for the VRP. Other researchers who contributed to ACOs that may be used to solve the VRP are Chen & Ting [19], Bin et al. [20], etc.

In this paper, we propose a novel pheromone deposition scheme that not only ensures that the pheromone concentration increase exponentially with time but also integrates the global feature and the local feature of the formed routes to facilitate faster convergence. We also propose a novel sequential constructive local search heuristic with greedy traits that may be utilized as the local search operator to yield fitter solutions. We also propose a novel two-way mutation scheme which ensures that sufficient diversity is maintained throughout computation. Hence, by virtue of this paper the novel features that we wish to incorporate in the standard ACO are: a novel Pheromone deposition scheme, a novel Local search scheme and a novel Mutation scheme; each of which can be adapted accordingly to solve a variety of combinatorial optimization problems. Hence, the proposed ACO_PLM can not only be used to solve the VRP efficiently but also as an efficient general combinatorial optimizer. The remainder of the paper can be organized as follows. The mathematical model for the VRP is discussed in Section 2. In Section 3 we discuss the novel ant system and its components in details. Some computational results are provided in Section 4 and finally, we conclude in Section 5.

2 The Vehicle Routing Problem (VRP)

The Vehicle Routing Problem (VRP) can be represented by a complete weighted directed graph where V represents the set of vertices and A represents the set of arcs. The vertex v_0 represents the depot and the remaining vertices represent the customers. With each arc $(i, j) \in A$ is associated a positive weight c_{ij} that represents the distance (or travel time/cost) between the two customers. Each customer has a non-negative demand d_i and each car has a capacity Q . We assume that the service time for each customer is zero. The aim is to minimize the net cost of all the vehicle routes with the following restrictions:

- Every customer is visited only once by a vehicle.
- All the routes begin and end at the depot.
- For each vehicle's route, the net demand should not exceed Q .

The VRP is an NP-hard combinatorial optimization problem that has been studied extensively over the past fifty years due to its paramount importance in the field of distribution management.

3 The Novel Ant System

3.1 Representation of the Solution

Figure 1 represents an example of the VRP. The corresponding tabular representation of this VRP is shown in Figure 2. The tabular representation is used for ease of computation. This is referred to as the *Route Table* (denoted by R_T). The node marked ‘0’ represents the *Depot* while the remaining nodes represent the customers. As can be inferred from Figure 1, each route begins and ends at the depot. For example, the route corresponding to Route ID₂ is actually $(v_0 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_0)$. Hence, the net distance that a vehicle has to traverse for a route: $(v_{n_1} \rightarrow v_{n_2} \rightarrow \dots \rightarrow v_{n_k})$ represented in the n^{th} row of a route table is given by:

$$D_n = dist(v_0, v_{n_1}) + \sum_{h=1}^{k-1} dist(v_{n_h}, v_{n_{(h+1)}}) + dist(v_{n_k}, v_0) \tag{1}$$

Hence, the fitness of this vehicle’s route, denoted by:

$$fitness(ID_n) = \frac{1}{D_n} \tag{2}$$

The overall distance that the vehicles need to cover for a given route table is given by the sum of the distances represented by the individual routes of the vehicles. The fitness of the complete route table, denoted by $fitness(R_T)$ is the inverse of this distance.

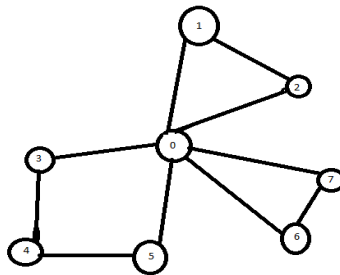


Fig. 1. An example of the VRP

| Route ID | List of Customers | | |
|----------|-------------------|---|---|
| 1 | 1 | 2 | |
| 2 | 3 | 4 | 5 |
| 3 | 6 | | 7 |

Fig. 2. Route Table

3.2 Key Components of the Ant Colony Optimization (ACO) Algorithm

Ants gather food by communicating with each other by means of a process referred to as stigmergy, wherein ants deposit on the ground a substance called pheromone forming in this way a pheromone trail. Ants can smell the pheromone and in probability choose the path with greater pheromone deposition. Using ACO, an individual ant simulates a vehicle, and its route is constructed by incrementally selecting customers until all customers have been visited. Initially, each ant starts at the depot and the set of customers included in its tour is empty. The ant selects the next customer to visit from the list of feasible locations and the storage capacity of the vehicle is updated before another customer is selected. The ant returns to the depot when the capacity constraint of the vehicle is met or when all customers are visited. The total distance L_T is computed as the objective function value for the complete route of the artificial ant. The ACO algorithm constructs a complete tour for the first ant prior to the second ant starting its tour. This continues until a predetermined number of ants M each construct a feasible route.

The key steps of the ACO algorithm are:

1) Initialization Phase: The various parameters are initialized. The different edges in the connected graph are initialized with their initial pheromone content. The ants are placed at the various nodes.

2) Route Construction Phase: The next node to which an ant located at node will move is determined by the probability (where K refers to the set of all the unvisited nodes in its neighbourhood and j refers to the particular unvisited node belonging to the set K that is under contention) as –

$$p_i^K(j) = \frac{\Gamma_{ij}^\alpha \times \eta_{ij}^\beta}{\sum_{j \in K} (\Gamma_{ij}^\alpha \times \eta_{ij}^\beta)} \tag{3}$$

In the above formula, Γ_{ij} refers to the pheromone concentration on the arc (i,j) : $\eta_{ij} = \frac{1}{d_{ij}}$ where d_{ij} refers to the distance between nodes i & j and α, β are constants. The ant travels to that unvisited node in its neighbourhood for which the transition probability value is the highest. This process is repeated for all the ants present in the system. During computation, a tabu list is maintained which updates

the list of nodes already visited by the ants. Once a node is visited, it cannot be visited again. Also, a check is maintained as to whether the capacity of the k^{th} ant (vehicle) has been exceeded or not. If exceeded, then the selected customer is assigned as the first customer in the route of the $(k+1)^{th}$ ant, and the aforementioned process is repeated again and again till the list of unvisited nodes becomes null.

3) Route Improvement/Mutation Phase: A route improvement heuristic/local search operator is applied to the obtained routes so as to facilitate faster convergence to the optimal solution. This may be accompanied by a mutation operator that may act on the obtained routes to promote diversity and ensure that the algorithm does not get trapped at local optima, which aren't as fit as the global best. This step isn't compulsory and is usually incorporated into the fabric of an ACO to facilitate convergence.

4) Pheromone Updating Phase: The pheromone concentration on each arc (i, j) is updated as:

$$\Gamma_{ij}(t) = (1 - \rho)\Gamma_{ij}(t-1) + \sum_{k=1}^M \Delta\Gamma_{ij}^k(t-1)$$

In the above formula, $\rho \in (0,1)$ represents the pheromone evaporation rate and $\Delta\Gamma_{ij}^k$ represents the pheromone deposited by the k^{th} ant (link) in a system of M ants.

3.3 Novel Pheromone Updating Rule

The traditional ACO proposed by Dorigo *et al.* utilized a uniform pheromone deposition rule, which in spite of its many merits had a major drawback as it cannot ensure that subsequent ants follow the same trajectory. However, an exponentially increasing time function ensures that subsequent ants close enough to a previously selected trial solution will follow the trajectory, as it can examine gradually thicker deposition of pheromones over the trajectory. Hence, theoretically the convergence characteristics of the ant system should improve as the deception probability is less. So, we have adopted an exponential pheromone deposition rule [21], which is described as follows –

We may rewrite the equation no. (3) as:

$$\begin{aligned} \Gamma_{ij}(t) - \Gamma_{ij}(t-1) &= (-\rho)\Gamma_{ij}(t-1) + \sum_{k=1}^M \Delta\Gamma_{ij}^k(t-1) \\ \Rightarrow \frac{d}{dt}(\Gamma_{ij}(t-1)) &= (-\rho)\Gamma_{ij}(t-1) + \sum_{k=1}^M \Delta\Gamma_{ij}^k(t-1) \\ \Rightarrow (D + \rho)(\Gamma_{ij}(t-1)) &= \sum_{k=1}^M \Delta\Gamma_{ij}^k(t-1) \end{aligned} \tag{5}$$

If we consider an exponentially increasing pheromone deposition rule:

$$\Delta\Gamma_{ij}^k(t-1) = F_k(1 - e^{-t/T}) \tag{6}$$

Then the complete solution of the above differential equation is given as:

$$\Gamma_{ij}(t) = [\Gamma_{ij}(0) - \sum_{k=1}^M \frac{F_k}{\rho} + \sum_{k=1}^M \frac{F_k e^{-1/T}}{\rho - 1/T}] \times e^{-\rho t} + \sum_{k=1}^M \frac{F_k}{\rho} - \sum_{k=1}^M \frac{F_k e^{-(t+1)/T}}{\rho - 1/T} \tag{7}$$

From the above equation number (6), it is quite ostensible that the value of $\Gamma_{ij}(t)$

converges rapidly to a steady value of $\sum_{k=1}^M \frac{F_k}{\rho}$ for positive values of ρ and T .

Hence, we utilize an exponentially increasing pheromone deposition rule in order to facilitate rapid convergence to the optimal solution.

The other novel component which we wish to incorporate in the proposed pheromone deposition scheme is a weighted rule that takes into consideration the overall fitness of the solution as well as the contribution of the current link to the solution. Hence, if L_t be the total length of a route and L_k be the contribution of the k^{th} ant (route adopted by the k^{th} vehicle), then,

$$F_k = \frac{Q_1}{L_t} + \frac{Q_2}{L_k} \quad \text{where } Q_1 \text{ \& } Q_2 \text{ are constants} \tag{8}$$

Since the strategy for updating the increased pheromone considered both the global feature and local feature of a solution, it can possibly ensure that the assigned increased pheromone is directly proportional to the quality of routes. Hence the more favorable the link/route is, the more is the pheromone increment allocated to it. Hence, more accurate directive information is provided for later search. Meanwhile, by adjusting the pheromone assigning method for the links of current optimal path automatically, the algorithm can facilitate more intensive search in the next cycle in a more favorable area, which helps in expanding its learning capacity from past searches.

3.4 Route Improvement/Mutation scheme

The two components of the Route Improvement/Mutation Scheme are:

1) Sequential Constructive Local Search heuristic (SCLS)

The SCLS heuristic aims to improve upon an existing route table by recursively operating on the constituent routes of the table, trying to make them fitter. It recursively searches for shorter sub-paths within a given traversal and combines them step-by-step to ensure that the derived solution is as fit as possible. This facilitates a faster convergence towards the optimal solution for the proposed algorithm. The SCLS is described in depth in Figure 3.

```

BEGIN SCLS ()
//  $R_T$  – Route Table
//  $R_T'$  – Derived Route Table
FOR  $i = 1; i \leq M$ 
     $p \leftarrow R_T(i)$ 
     $n(p) \leftarrow$  Number of customers in the route  $p$ 
    // All the derived routes are stored in matrix  $Z$ 
     $l = 1;$ 
    FOR  $j = 1; j \leq n(p)$ 
         $Z(j, l) = p(j);$ 
        while( $l \leq n(p)$ )
            • Search for  $x_{right}$ ; //  $x_{right}$  is the first element to the right of the current element
              which has not occurred in the route.
            • Search for  $x_{left}$ ; //  $x_{left}$  is the first element to the left of the current element
              which has not occurred in the route.
             $l = l + 1;$ 
            if (  $dist(Z(j, l-1), x_{left}) \geq dist(Z(j, l-1), x_{right})$  )
                 $Z(j, l) = x_{left};$ 
            else
                 $Z(j, l) = x_{right};$ 
            end-if
        END  $j$ -loop
    // Store the fittest entity of  $Z$  in  $p'$ .
    if ( $fitness(p') \geq fitness(p)$ )
         $R_T'(i) = p';$ 
    else
         $R_T'(i) = p;$ 
    end
     $i = i + 1;$ 
END  $i$ -loop
if ( $fitness(R_T') \geq fitness(R_T)$ )
     $R_T = R_T';$ 
    Call SCLS();
else
    Abort operation.
END SCLS()

```

Fig. 3. Pseudo-code for the Sequential Local Search heuristic

2) Mutation Scheme

The mutation scheme that we propose in this paper is two-fold, i.e. there are two kinds of mutation that a route table can undergo:

Inter-route mutation – In this mutation scheme, two routes are selected from the list of M routes, and then any two cities belonging to the two routes are exchanged with each other. But, the drawback of this scheme is that the resulting route table so obtained

may not be a valid one, i.e. the maximum capacity of a vehicle may be exceeded as a result of the 2-opt exchange. Hence, every pair of cities belonging to the two routes are considered and 2-opt exchange is applied to them until the resulting route table is a valid one. In case no valid route table can be obtained, the inter-route mutation operation is aborted and the original route table remains unaltered.

Intra-route mutation – This mutation scheme suggests that any two customers (nodes) that fall in the route of the same vehicle (ant) are selected randomly and exchanged. As the net demand of the customers present in the route remains unaltered, the mutant so obtained will definitely represent a valid route table.

We must note that during mutation, a solution may undergo either one or both of the afore-mentioned mutations. As it is desirable that the diversity of the population is initially large and it should decrease with time to promote convergence, the mutation rate is adapted accordingly. If the maximum number of iterations is T , then the mutation probabilities for undergoing Inter-route mutation and Intra-route mutation in the t^{th} iteration are given by (where, μ_{inter}^{min} , μ_{inter}^{max} , μ_{intra}^{min} & μ_{intra}^{max} are constants) :

$$\mu_{inter}(t) = \mu_{inter}^{min} + (\mu_{inter}^{max} - \mu_{inter}^{min})\left(\frac{t}{T}\right) \tag{9}$$

$$\mu_{intra}(t) = \mu_{intra}^{min} + (\mu_{intra}^{max} - \mu_{intra}^{min})\left(\frac{t}{T}\right) \tag{10}$$

3.5 Algorithm Flowchart

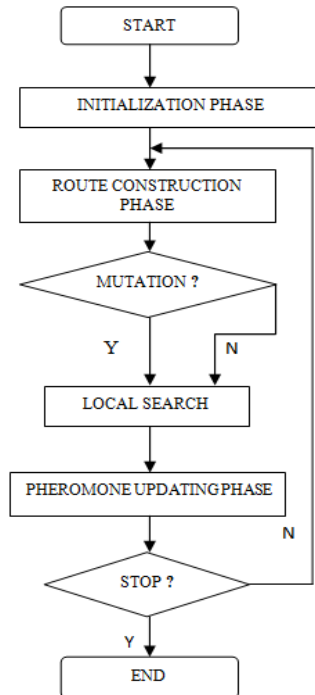


Fig. 4. ACO_PLM Flowchart

4 Results and Discussion

In our tests on the VRP problem instances, the proposed ACO_PLM algorithm is coded in MATLAB and the experiments are executed on a Pentium P-IV 3.0 GHz PC with 512MB memory. The twelve VRP instances on which we've performed our tests can be downloaded from the OR library [22]. The optimal solution for each instance is also listed in [22]. We have compared our algorithm with the Improved ACO (IACO) as proposed by Bin et al. [20], the Improved genetic algorithm(IGA) for the VRP proposed by Honglin & Jijun [23] and the parallel tabu search algorithm (RR-PTS) by Rego and Roucairol [24]. The values assigned to the various parameters of the ACO_PLM when tested upon all the instances are stated in Table No. 1 and the results are compiled in Table No. 2.

The parameter values defined in Table 1 have been determined by careful analysis as well as trial and error. The values of α , β , Q_1 , Q_2 have been determined over several trials of the ACO_PLM on the given 12 instances. As can be seen from the afore-mentioned values, we allow the predominance of Intra-route mutation over Inter-route mutation because it may be found by virtue of analysis that the SCLS() is effective in obtaining extremely fit solutions from original solutions that may be less fit or quite unfit. Hence, if an otherwise fit path is affected adversely by virtue of the intra-route mutation then the local search heuristic is able to annul the negative effect by virtue of its characteristic extensive search property. But in case of inter-route mutation, customers that may be geographically very distant from each other may be coupled into a vehicle's route and this may not augur well with the SCLS() due to the presence of one or more isolated nodes, which may make it impossible to derive fit route(s) starting from the original path traversal.

For each instance of the VRP we have stated the value of the best known solution. We have stated the best, average and worst result for all the four algorithms in Table II. The afore-mentioned results have been obtained over 50 runs of each algorithm on all the problem instances. As evident from the table, the proposed ACO_PLM clearly outperforms the IGA and the RR-PTS. Its performance is also found to be more consistent than that of the IACO as inferred from over fifty runs where both the algorithms were continuously compared with each other.

Table 1. The optimal values of the parameters of ACO_PLM

| α | β | Q_1 | Q_2 | μ_{inter}^{max} | μ_{inter}^{min} | μ_{intra}^{max} | μ_{intra}^{min} | T | ρ |
|----------|---------|-------|-------|---------------------|---------------------|---------------------|---------------------|-----|--------|
| 2.3 | 1.1 | 1200 | 450 | 0.98 | 0.80 | 0.95 | 0.65 | 200 | 0.2 |

Table 2. Comparison of the results over 50 runs obtained using ACO_PLM, IACO, IGA and RR-PTS for the twelve VRP instances

| <i>Algorithm</i> <i>Problems</i> | <i>Optimal Solution</i> | <i>Statistics</i> | <i>ACO_PLM</i> | <i>IACO</i> | <i>IGA</i> | <i>RR PTS</i> |
|-------------------------------------|-------------------------|-------------------|----------------|-------------|------------|---------------|
| C1 | 524.61 | Best | 524.61 | 524.61 | 524.61 | 524.61 |
| | | Average | 524.61 | 524.61 | 524.61 | 524.61 |
| | | Worst | 524.61 | 524.61 | 524.61 | 524.61 |
| C2 | 835.26 | Best | 835.26 | 835.26 | 838.49 | 843.6 |
| | | Average | 838.88 | 848.85 | 852.36 | 858.20 |
| | | Worst | 844.56 | 859.30 | 861.12 | 871.23 |
| C3 | 826.14 | Best | 828.28 | 830.00 | 842.13 | 842.13 |
| | | Average | 835.12 | 844.32 | 850.34 | 867.00 |
| | | Worst | 844.86 | 861.12 | 872.34 | 870.24 |
| C4 | 1028.42 | Best | 1028.42 | 1028.42 | 1044.89 | 1046.55 |
| | | Average | 1044.12 | 1042.52 | 1058.62 | 1055.32 |
| | | Worst | 1053.12 | 1067.10 | 1087.3 | 1080.92 |
| C5 | 1291.45 | Best | 1291.45 | 1305.56 | 1335.36 | 1310.21 |
| | | Average | 1310.11 | 1321.91 | 1362.37 | 1340.07 |
| | | Worst | 1322.41 | 1344.41 | 1387.85 | 1377.29 |
| C6 | 555.43 | Best | 556.23 | 555.43 | 555.43 | 556.78 |
| | | Average | 561.23 | 560.14 | 564.18 | 566.78 |
| | | Worst | 570.82 | 568.89 | 577.49 | 581.11 |
| C7 | 909.68 | Best | 909.68 | 909.68 | 909.68 | 912.37 |
| | | Average | 915.22 | 919.10 | 931.07 | 930.06 |
| | | Worst | 932.26 | 942.29 | 950.1 | 944.23 |
| C8 | 865.94 | Best | 865.94 | 865.94 | 876.52 | 882.32 |
| | | Average | 872.23 | 871.52 | 888.87 | 892.34 |
| | | Worst | 890.11 | 888.89 | 901.06 | 910.23 |
| C9 | 1162.55 | Best | 1162.55 | 1162.55 | 1188.00 | 1190.11 |
| | | Average | 1190.23 | 1194.87 | 1222.24 | 1240.12 |
| | | Worst | 1208.20 | 1228.90 | 1253.31 | 1282.12 |
| C10 | 1395.85 | Best | 1395.85 | 1395.85 | 1430.12 | 1439.07 |
| | | Average | 1413.44 | 1412.92 | 1472.33 | 1487.78 |
| | | Worst | 1433.68 | 1433.68 | 1504.12 | 1520.04 |
| C11 | 1042.11 | Best | 1042.11 | 1042.11 | 1051.71 | 1055.65 |
| | | Average | 1044.16 | 1048.12 | 1059.13 | 1067.88 |
| | | Worst | 1052.20 | 1056.26 | 1077.33 | 1080.90 |
| C12 | 819.56 | Best | 819.56 | 819.56 | 833.31 | 833.31 |
| | | Average | 824.56 | 823.66 | 840.61 | 843.32 |
| | | Worst | 836.78 | 842.51 | 850.04 | 864.43 |

5 Conclusion

Since a delivery route can comprise of absolutely any combination of customers, the VRP is an NP-hard combinatorial optimization problem. In this paper we have used a modified version of the ACO in order to solve the VRP efficiently. Here we have mainly looked for better convergence speed and accuracy. A novel pheromone deposition scheme has been proposed that not only ensures faster convergence but also ensures that the global feature is integrated with the local feature. A novel sequential constructive local search operator is proposed which paves way for faster convergence as well as fitter solutions. At the same time to maintain better convergence speed we have introduced some new mutation techniques which maintain diversity and prevent the algorithm from converging at local optima. We have compared its performance with that of three other relevant algorithms and from the results obtained, we find that the proposed algorithm performs more efficiently than the rest. Furthermore, as the proposed algorithm is not studded with features that are only specific to the VRP, it can be treated as a general combinatorial optimizer. Hence, we may infer that by virtue of this algorithm, we have maximized the speed

and the accuracy of the search process. Further research can include modification of the ACO_PLM and its adaption to solve other versions of the VRP that also consider time windows, more depots, etc.

References

- [1] Christofides, N., Mingozzi, A., Toth, P.: The vehicle routing problem. *Combinatorial optimization*. *Combinatorial Optimization* 11, 315–338 (1979)
- [2] Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12, 568–581 (1964)
- [3] Taillard, R.E.: Parallel iterative search methods for vehicle routing problems. *Networks* 23, 661–673 (1993)
- [4] Chiang, W.C., Russell, R.: Simulated annealing meta-heuristics for the vehicle routing problem with time windows. *Annals of Operations Research* 93, 3–27 (1996)
- [5] Osman, I.H.: Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* 41, 421–451 (1993)
- [6] Berger, J., Barkaoui, M.: An Improved Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows. In: *International ICSC Symposium on Computational Intelligence, Part of the International ICSC Congress on Intelligent Systems and Applications (ISA 2000)*, University of Wollongong, Wollongong (2000)
- [7] Tan, K.C., Lee, L.H., Ou, K.: Hybrid Genetic Algorithms in Solving Vehicle Routing Problems with Time Window Constraints. *Asia-Pacific Journal of Operational Research* 18, 121–130 (2001)
- [8] Osman, M.S., Abo-Sinna, M.A., Mousa, A.A.: An effective genetic algorithm approach to multiobjective routing problems (morps). *Applied Mathematics and Computation* 163, 769–781 (2005)
- [9] Marinakis, Y., Marinaki, M.: A Hybrid Multi-Swarm Particle Swarm Optimization algorithm for the Vehicle Routing Problem. *Computers and Operations Research* 37(3), 432–442 (2010)
- [10] Ai, J., Kachitvichyanukul, V.: A Study on Adaptive Particle Swarm Optimization for Solving Vehicle Routing Problems. In: *The 9th Asia Pacific Industrial Engineering and Management Systems Conference* (2008)
- [11] Reimann, M., Stummer, M., Doerner, K.: A savings based ant system for the vehicle routing problem. In: Langdon, W.B., et al. (eds.) *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*. Morgan Kaufmann, San Francisco (2002)
- [12] Stutzle, T., Dorigo, M.: ACO algorithms for the traveling salesman problem. In: *Evolutionary Algorithms in Engineering and Computer Science*, John Wiley and Sons (1999)
- [13] Reinelt, G.: *The traveling salesman: computational solutions for TSP applications*. LNCS, vol. 840. Springer (1994)
- [14] McCormich, S.T., Pinedo, M.L., Shenker, S., Wolf, B.: Sequencing in an assembly line with blocking to minimize cycle time. *Operations Research* 37, 925–936 (1989)
- [15] Leisten, R.: Flowshop sequencing problems with limited buffer storage. *International Journal of Production Research* 28, 2085–2100 (1994)
- [16] Kuntz, P., Layzell, P., Snyers, D.: A colony of ant-like agents for partitioning in VLSI technology. In: *Husbands, P., Harvey, I. (eds.) Proc. of 4th European Conference on Artificial Life*, pp. 417–424. MIT Press, Cambridge (1997)

- [17] Bonabeau, E., Dorigo, M., Theraulaz, G.: *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press (1999)
- [18] Bullnheimer, B., Hartl, R.F., Strauss, C.: Applying the ant system to the vehicle routing problem. In: *Second Metaheuristics International Conference, MIC 1997*, Sophia-Antipolis, France (1997)
- [19] Chen, C.H., Ting, C.J.: An improved ant colony system algorithm for the vehicle routing problem. *Journal of the Chinese Institute of Industrial Engineers* 23(2), 115–126 (2006)
- [20] Bin, Y., Zhong-Zen, Y., Baozhen, Y.: An Improved ant colony optimization for the Vehicle Routing Problem. *European Journal of Operational Research* 196, 171–176 (2009)
- [21] Abraham, A., Konar, A., Samal, N.R., Das, S.: Stability Analysis of the Ant System Dynamics with Non-uniform Pheromone Deposition Rules. In: *Proc. IEEE Congress on Evolutionary Computation*, pp. 1103–1108 (2007)
- [22] Beasley, J.E.: OR-Library: distributing test problems by electronic mail. *Journal of the Operational Research Society* 41, 1069–1072 (1990)
- [23] Honglin, Y., Jijun, Y.: An Improved Genetic Algorithm for the Vehicle Routing Problem (2002)
- [24] Rego, C., Roucairol, C.: A parallel tabu search algorithm using ejection chains for the vehicle routing problem. In: *Meta-Heuristics*, pp. 661–675. Springer US (1996)

Implementation of Fractional Order PID Controller for Three Interacting Tank Process Optimally Tuned Using Bee Colony Optimization

U. Sabura Banu

EIE Department,
BS Abdur Rahman University,
Vandalur, Chennai – 48
sabura_banu@rediffmail.com

Abstract. The proposed work demonstrates the application of Bee Colony Optimization (BCO) technique for the tuning of Fractional Order Proportional-Integral-Derivative (FOPID) controller for Three Interacting Tank system. FOPID controller parameters are composed of proportionality constant, integral constant, integral order, derivative constant and derivative order. Grunwald-Letnikov definition is used for the defining the derivative controller and Oustaloup's filter technique is used for the approximation of the function. Tuning FOPID controller parameters is more complicated as it involves a five dimensional search. Tuning is effected using an evolutionary optimization technique, the bee colony optimization so as to minimize the Integral Time Absolute Error (ITAE). The proposed technique is used to control three interacting tank process. The proposed FOPID controller tuned using Bee colony optimization technique may serve as an alternative for the tuning of the fractional order controllers.

1 Introduction

Fractional calculus is thriving for the past two decades with the research progress in the field of chaos. Fractional order modeling and control design for dynamical system are still in introductory stage. The basic concepts of fractional order calculus is dealt with by Gement, Méhauté, Oustaloup and Podlubny [1-4]. Fractional calculus finds its role in varied fields like Chaos, fractals, biology, electronics, communication, digital signal processing, dynamical system modelling and control etc. [5-8]. Design of fractional order dynamical models and controller have gained popularity in recent past with the development of fractional calculus [9-13,16]. Fractional order PID controller was conceptualized by Podlubny and effectiveness of the proposed work was demonstrated. Realization of the FOPID controller. Various studies have been carried out in the frequency domain approach, pole distribution of characteristic equation, pole placement technique evolved [14]. Integrating a fractional component in a integer order controller will result in FOPID. Till date PID controllers are termed as workhorse in automation of many industrial applications. The only change to be made is the introduction of fractional order to the integral and derivative component. Tuning

the fractional order PID controller parameter is complicated because of the five dimensional search requirements. Evolutionary optimization techniques like Genetic algorithm, Particle Swarm optimization, Bacterial foraging, differential evolution, Bee colony optimization etc. are gaining popularity in the tuning of the controller parameters. Bee colony Optimization is a swarm intelligent technique used to find a global optimal solution for any real-world problems. The BCO algorithm is based on the model that is obtained from the communicative behavior of the honey bees. Bee colony optimization algorithm is inspired by the behavior of a honey bee colony in nectar collection. This biologically inspired approach is currently being employed to solve continuous optimization problems. The bee colony optimization algorithm is inspired by the behavior of a honey bee colony in nectar collection[15]. This biologically inspired approach is currently being employed to solve continuous optimization problems, training neural networks, mechanical and electronic components design optimization, combinatorial optimization problems etc. The make span of the solution is analogous to the profitability of the food source in terms of distance and sweetness of the nectar. Hence, the shorter the make span, the higher the profitability of the solution path. Once a feasible solution is found, each bee will return to the hive to perform a waggle dance. The waggle dance will be represented by a list of "elite solutions", from which other bees can choose to follow another bee's path. Bees with a better make span will have a higher probability of adding its path to the list of "elite solutions", promoting a convergence to an optimal solution. In the proposed work, the fractional order PID controller parameters is brought out by using the optimal values obtained from BCO.

2 Three Interacting Tank Process Description

The level control of three interacting tanks involves complex design and implementation procedure, as the response of each tank depends on the response of other tanks. Moreover the process is non-linear. Using classical control technique, the level control becomes complicated. The plant has three interacting tanks interconnected by manual control valves. The flow f_{in1} to the first tank and the flow f_{in2} to the third tank are the plant inputs. The levels of the three interacting tanks are the outputs of the plant. Thus the plant is a two inputs and three output systems. The process liquid is pumped to the first interacting tank from the sump by pump 1 through the control valve 1 and the input flow to the first interacting tank is f_{in1} . The process liquid is pumped to the third interacting tank from the sump by pump 2 through the control valve 2 and this input flow to the third interacting tank is f_{in2} . The three levels in the three interacting tank are measured using differential pressure transmitters. The three interacting tanks are interconnected through manual control valves. The inputs are voltages u_1 converted to 4-20mA current that causes the actuation of the control valve and adjusts the inflow f_{in1} into the first interacting tank and voltage u_2 converted to 4-20mA current that causes the actuation of the control valve and adjusts the inflow f_{in2} to the third interacting tank. In the present work, u_2 is kept constant and u_1 is controlled. The setup diagram of the three interacting tanks is shown in figure 1.



Fig. 1. Hardware Set-up of Three Interacting Tank Level Process

The setup consists of three cylindrical process tanks, overhead sump, submersible pump, three Differential Pressure Transmitter (DPT), rotameter, control valve and interfacing card. The process tank is cylindrical and made of a transparent glass. Provisions for water inflow and outflow are provided at the top and bottom of the tank respectively. A pump is used for discharging the liquid from the storage tank to Tank 1. The inflow to the Tank 1 is maintained by a control valve. Differential Pressure Transmitter is used for liquid level measurement. In this open vessel process, tank pressure is given to the high-pressure side of the transmitter and the low-pressure side is vented to the atmosphere. Rotameter is used for the monitoring of the level. Gate valves one each at the outflow of the tank 1, 2 and 3 are also provided to maintain the level of water in the tanks. Clockwise rotation ensures the closure of the valve, thus stopping the flow of liquid and vice versa. A 25-pin male connector is used here to interface the hardware setup with the PC. The electrical output generated from the potentiometer is first converted into a digital value before applying it to the computer. The process parameters of the three interacting tank set in tabulated in Table 1.

Table 1. Process Parameters of the three interacting tank

| Sl. No. | Process parameters | Values |
|---------|--|----------------------------|
| 1. | Area of the Tank, A_i | 615.7522cm^2 |
| 2. | Area of the connecting pipes, a_{ij} | 5.0671cm^2 |
| 3. | Valve ratio | |
| | β_{12} | 0.9 |
| | β_{23} | 0.8 |
| | β_3 | 0.3 |
| 4. | Pump Gain, K_i | $75\text{cm}^3/\text{V.s}$ |

3 Input-Output Characteristics

The three interacting process has two manipulated variable the inflow to the tank 1 (u_1) and tank 3 (u_2) regulated by the control valve. The output of the process are the levels (h_1, h_2, h_3) of the tanks 1, 2 and 3 respectively. The inflow u_2 is maintained constant at 0.5V and the open loop response of the three interacting tanks is obtained by varying the voltage u_1 from 0.5V to 7.5V. The steady state I/O data are tabulated in Table 2. The I/O characteristics of the three interacting tanks for varying values of u_1 are shown in figure. 2. In the present work, the inflow u_1 is considered as the manipulated variable and the height of the third tank h_3 is considered as the output of the process.

Table 2. I/O Data obtained from the Lab Scale setup

| F_{in1} | h_1 | h_2 | h_3 | F_{in1} | h_1 | h_2 | h_3 |
|-----------|-------|--------|--------|-----------|-------|--------|--------|
| 0.500 | 1.310 | 1.280 | 1.240 | 4.50 | 37.35 | 34.560 | 31.020 |
| 1.000 | 3.105 | 2.968 | 2.792 | 5.00 | 45.38 | 41.860 | 37.560 |
| 1.250 | 4.288 | 4.077 | 3.800 | 5.50 | 54.11 | 49.995 | 44.665 |
| 1.500 | 5.660 | 5.350 | 4.960 | 6.00 | 63.66 | 58.745 | 52.425 |
| 2.000 | 9.001 | 8.465 | 7.753 | 6.50 | 73.90 | 68.100 | 60.700 |
| 2.500 | 13.12 | 12.260 | 11.169 | 6.75 | 79.25 | 73.180 | 65.135 |
| 3.000 | 18.01 | 16.770 | 15.203 | 7.00 | 85.10 | 78.400 | 69.700 |
| 3.500 | 23.67 | 22.000 | 19.850 | 7.50 | 96.99 | 89.260 | 79.420 |
| 4.000 | 30.13 | 27.914 | 25.137 | | | | |

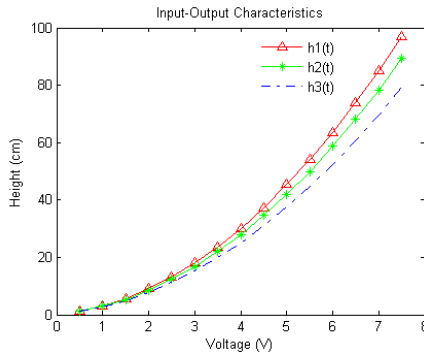


Fig. 2. I/O characteristics of the three interacting tanks for varying values of u_1

4 Fractional Calculus

The differintegral operator, ${}_aD_t^q$, is a combined differentiation-integration operator commonly used in fractional calculus.

This operator is a notation for taking both the fractional derivative and the fractional integral in a single expression and is defined by

$${}_a D_t^q = \begin{cases} \frac{d^q}{dt^q} & q > 0 \\ 1 & q = 0 \\ \int_a^t (d\tau)^{-q} & q < 0 \end{cases}$$

where q is the fractional order which can be a complex number and a and t are the limits of the operation. There are some definitions for fractional derivatives. The commonly used definitions are Grunwald-Letnikov, Riemann-Liouville and Caputo definitions (Podlubny1999b). The Grunwald-Letnikov definition is given by

$${}_a D_t^q = \frac{d^q f(t)}{d(t-a)^q} = \lim_{N \rightarrow \infty} \left[\frac{t-a}{N} \right]^q \sum_{j=0}^{N-1} (-1)^j \binom{q}{j} f\left(t-j \left[\frac{t-a}{N} \right]\right)$$

The Riemann-Liouville definition is the simplest and easiest definition to use. This definition is given by

$${}_a D_t^q f(t) = \frac{d^q f(t)}{d(t-a)^q} = \frac{1}{\Gamma(n-q)} \frac{d^n}{dt^n} \int_0^t (t-\tau)^{n-q-1} f(\tau) d\tau$$

Where n is the first integer which is not less than q i.e. $n-1 \leq q < n$ and Γ is the Gamma function.

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt$$

For functions $f(t)$ having n continuous derivatives for $t \geq 0$ where $n-1 \leq q < n$, the Grunwald-Letnikov and the Riemann-Liouville definitions are equivalent. The Laplace transforms of the Riemann-Liouville fractional integral and derivative are given as follows:

$$L\{ {}_a D_t^q f(t) \} = s^q F(s) - \sum_{k=0}^{n-1} s^k {}_0 D_t^{q-k-1} f(0); \quad n-1 < q < n$$

The Riemann-Liouville fractional derivative appears unsuitable to be treated by the Laplace transform technique because it requires the knowledge of the non-integer order derivatives of the function at $t=0$. This problem does not exist in the Caputo definition that is sometimes referred as smooth fractional derivative in literature. This definition of derivative is defined by

$${}_a D_t^q f(t) = \begin{cases} \frac{1}{\Gamma(m-q)} \int_0^t \frac{f^{(m)}(\tau)}{(t-\tau)^{q+1-m}} d\tau; & m-1 < q < m \\ \frac{d^m}{dt^m} f(t) & ; \quad q=m \end{cases}$$

Where m is the first integer larger than q . It is found that the equations with Riemann-Liouville operators are equivalent to those with Caputo operators by homogeneous initial conditions assumption. The Laplace transform of the Caputo fractional derivative is

$$L\{ {}_n D_t^q f(t) \} = s^q F(s) - \sum_0^{n-1} s^{q-k-1} f^{(k)}(0); \quad n-1 < q < n$$

Contrary to the Laplace transform of the Riemann-Liouville fractional derivative, only integer order derivatives of function f are appeared in the Laplace transform of the Caputo fractional derivative. For zero initial conditions, previous equation reduces to

$$L\{ {}_0D_t^q f(t) \} = s^q F(s)$$

The numerical simulation of a fractional differential equation is not simple as that of an ordinary differential equation. Since fractional order differential equations do not have exact analytic solutions, approximations and numerical techniques are used. The approximation method, Oustaloup filter is given by

$$s^q = k \prod_{n=1}^N \frac{1 + \frac{s}{\omega_{zn}}}{1 + \frac{s}{\omega_{pn}}} \quad ; \quad q > 0$$

The approximation is valid in the frequency range $[\omega_l, \omega_h]$; gain k is adjusted so that the approximation shall have unit gain at 1 rad/sec; the number of poles and zeros N is chosen beforehand (low values resulting in simpler approximations but also causing the appearance of a ripple in both gain and phase behaviours); frequencies of poles and zeros are given by

$$\alpha = \left(\frac{\omega_h}{\omega_l} \right)^{\frac{q}{N}}$$

$$\eta = \left(\frac{\omega_h}{\omega_l} \right)^{\frac{1-q}{N}}$$

$$\omega_{zn} = \omega_{p,n-1} \eta, \quad n=2, \dots, N$$

$$\omega_{pn} = \omega_{z,n-1} \alpha, \quad n=1, \dots, N.$$

5 Bee Colony Optimization Technique

The ways of communication of the bee for find the food sources is very interesting. In the bee hive, the worker bees are responsible for food collection. The worker bees in a honey bee colony are grouped as food-storer, scout and forager. The food collection is organized by the colony by recruiting bees for different jobs. The recruitment is managed by the forager bees which can perform dances to communicate with their fellow bees inside the hive and recruit them. The scouts are sent to different directions in search of honey. Each bee visits a number of flowers. The floral finding are communicated to the other bees through dancing. At the entrance of the hive is an area called the dance-floor, where dancing takes place.

Different types of dances have been identified:

Waggle Dance

It is an advertisement for the food source of the dancer. Another forager can leave her food source and watch out for a well advertised food source. A forager randomly

follows dances of multiple recruiting foragers and seems to respond randomly as well. Especially a bee does not compare several dances. A dance does not seem to contain any information that helps to choose a food source.

Tremble Dance

The foragers are more likely to perform the tremble dance if they have to wait long for a food-storer bee to unload their nectar after their arrival at hive. Foragers perform the tremble dance on the dance-floor and in the brood nest as well, whereas the waggle dance is limited to the dance-floor. So maybe bees in the hive are addressed, too. According to Seeley worker bees in the hive are ordered by the tremble dancers to give up their jobs and to unload nectar.

The orientation of the bee along with the frequency of the vibrations indicates the direction and the distance of the flowers from the hive. The dance is observed by the other bees which will find the location and the quantity of the food source. All the worker bees, who finds flower will dance, but majority of the worker bees will follow the elite bee-the bee which had located a better and nearer food source than the remaining bees.

6 Bee Colony Optimized FOPID Tuning for Three Interacting Tank Process

Figure 3 shows the block diagram of the FOPID controller optimized using bee colony technique. The Bees algorithm is developed to obtain the optimal controller parameters such as K_p , K_I , K_D , λ and μ Then it is implemented in FOPID controller to obtain the response.

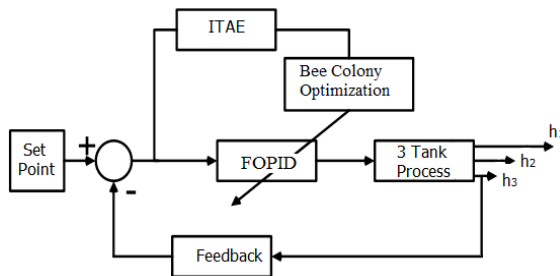


Fig. 3. Block diagram of FOPID controller tuned Three Interacting Tank Process

6.1 Performance Index

In this section, the feedback controller design is formulated as an optimization problem and the solution is sort through steps of Ant Colony Optimization (ACO). The technique uses ACO to tune the PID parameters online for a minimum ITAE for each region separately. Due to the variety of PID control law permutations, it is

necessary to specify a minimum set of attributes that is PID controller is assumed to be of non-interacting form as defined below:

$$G_c(s) = K_p + K_i \frac{1}{s^\lambda} + K_d s^\mu$$

By suitable transformation of the parameters this form is converted to interacting form. Minimizing the following error criteria generates the controller parameter

$$ITAE = \int_0^T |r(t) - y(t)| t dt$$

where r(t) = reference input, y(t) = measured variable

At first, the bee, i.e. the PID parameters are randomly initialized. The fitness function is defined as 1/ITAE. Smaller the fitness function, the better performance of the system response with the specified PID parameters.

6.2 Algorithm: Bee Colony Optimization for FO-PID Controller Tuning

1. Initialize the population of solutions $x_{i,j}$
2. Evaluate the population
3. Cycle =1
4. Repeat
5. Produce new solutions (food source positions) $v_{i,j}$ in the neighbourhood of $x_{i,j}$ for the employed bees using the formula

$$v_{i,j} = x_{i,j} + \phi_{ij}(x_{i,j} - x_{k,j})$$

where k is the solution in the neighbourhood of I, ϕ is a random number and evaluate them

6. Apply the greedy selection process between x_i and v_i
7. Calculate the probability values P_i for the solutions x_i by means of their fitness values using

$$P_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i}$$

The fitness values of solutions are calculated as

$$fit_i = \begin{cases} \frac{1}{1 + f_i} & \text{if } f_i \geq 0 \\ 1 + \text{abs}(f_i) & \text{if } f_i < 0 \end{cases}$$

Normalize P_i values into [0,1]

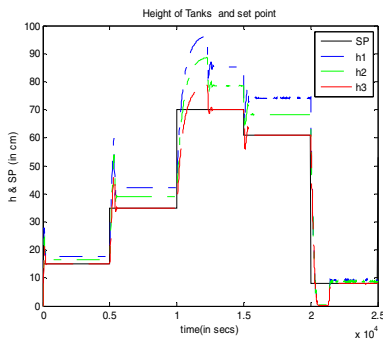
8. Produce the new solutions (new positions) v_i for the onlookers from the solutions x_i , selected depending on P_i and evaluate them
9. Apply the greedy selection process for the onlookers between v_i and x_i
10. Determine the abandoned solution (source) if exists and replace it with a new randomly produced solution x_i for the scout using $x_{ij} = \min_j + \text{rand}(0,1) * (\max_j - \min_j)$

11. Memorize the best food source position (solution) achieved so far
12. Cycle = cycle +1, Until cycle = Maximum cycle Number

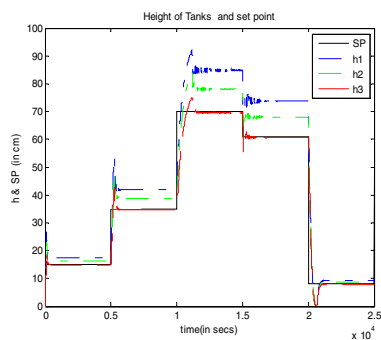
The termination criterion can be in two was: either by ending the program when objective function value reaches a reasonably low value or after a finite number of iterative steps. In the present case, the program was terminated after 45 iterations.

7 Comparison of BCO-IOPID Controller with BCO-FOPID Controller

The BCO algorithm was implemented by developing a dedicated software using MATLAB. Figure 4 shows the heights h_1 , h_2 and h_3 of the three interacting tanks for the set values fixed are 15cm, 35cm, 70cm, 61cm and 8cm at an interval of 5000s. Figure 5 shows the response for the regulatory problem with set point fixed at 50cm and disturbance applied at an interval of 2500s using an integer and fractional order controller. Figure 6 shows the response for the servo regulatory problem for the three interacting tank process



4.a. Integer order PID controller



4.b. Fractional order PID controller

Fig. 4. Height changes in all the three tanks for servo change with set point change of 15cm, 35cm,70cm, 61cm and 8 cm at an interval of 5000 secs

Table 3. Performance indices for the IOPID and FOPID controller for three interacting tank process

| Control | Performance Indices | | | Performance Indices | | |
|------------------------------|--------------------------------------|-------|--------------------|--|-------|--------------------|
| | IAE | ISE | ITAE | IAE | ISE | ITAE |
| Integer order PID controller | $2.7079 + \frac{0.0938}{s} + 0.075s$ | | | Fractional Order PID controller $0.8242 + \frac{0.9978}{s^{0.3155}} + 0.4989s^{0.2225}$ | | |
| Servo | 50380 | 71930 | $5.943 \cdot 10^8$ | 38876 | 50290 | $4.897 \cdot 10^8$ |
| Regulatory | 32460 | 43850 | $1.742 \cdot 10^8$ | 26780 | 32123 | $1.358 \cdot 10^8$ |
| Servo-Regulatory | 53890 | 74280 | $7.085 \cdot 10^8$ | 37484 | 45761 | $4.183 \cdot 10^8$ |

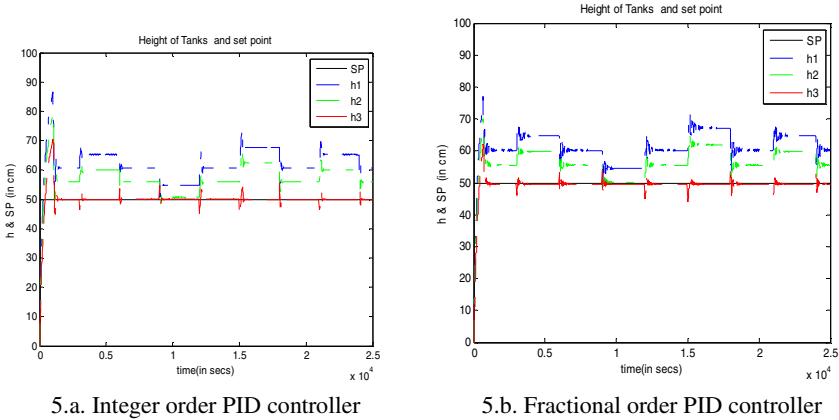


Fig. 5. Height changes in all the three tanks for regulatory change with set point fixed at 50cm and disturbance applied at an interval of 2500 secs

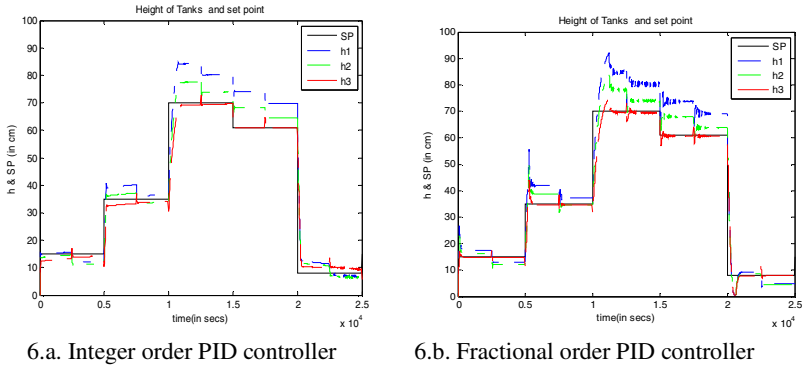


Fig. 6. Height changes in all the three tanks for servo-regulatory change for set point change of 15cm, 35cm,70cm, 61cm and 8 cm at an interval of 5000 secs and disturbances applied at 2500s, 7500s,12500s, 17,500s and 22500s respectively

8 Conclusion

Three interacting tank process control is effected using Bee colony optimization technique based Integer order and Fractional Order PID controller. The performance selected is Integral Time Absolute Error. The integer order PID controller involves optimization of three parameters K_p , K_i and K_d whereas the fractional order controller involves two more additional parameters λ and μ increasing the number of controller parameters as five. Applying bee colony optimization, the integer order parameters are found out. The work can be extended by implementation of other swarm intelligence techniques like particle swarm optimization, ant colony optimization, etc. On investigation of the performance indices, it can be observed that the IAE, ISE and ITAE for servo and servo-regulatory problem for a fractional order controller is less

compared to the integer order controller. On analysis, it can be inferred that the fractional order controller can be implemented for the control of a nonlinear process. The scope of the future work is to implement other swarm intelligence techniques like particle swarm optimization, bacterial foraging, fish schooling and evolutionary optimization techniques like genetic algorithm, differential evolution etc. and the performance can be computed.

References

1. Gement, A.: On fractional differentials. *Proc. Philosophical Magazine* 25, 540–549 (1938)
2. Méhauté, A.L.: *Fractal Geometries: Theory and Applications*. Penton Press (1991)
3. Oustaloup, A.: *La Commande CRONE: Commande Robust Order Non Integer*. Hermes (1991)
4. Podlubny, I.: *Fractional Differential Equations*. Academic Press, San Diego (1999)
5. Tenreiro Machado, J.A.: System modelling and control through fractional-order algorithms. *FCAA – Journal of Fractional Calculus and Ap. Analysis* 4, 47–66 (2001)
6. Torvik, P.J., Bagley, R.L.: On the appearance of the fractional derivative in the behaviour of real materials. *ASME Journal of Applied Mechanics* 51, 294–298 (1984)
7. Vinagre, B.M., Petras, I., Podlubny, I., Chen, Y.Q.: Using fractional order adjustment rules and fractional order reference models in model-reference adaptive control. *Nonlinear Dynamics* 29, 269–279 (2002)
8. Westerlund, S.: *Dead Matter Has Memory*. Causal Consulting Kalmar, Sweden (2002)
9. Astrom, K., Hagglund, T.: *PID Controllers: Theory, Design and Tuning*. Instrument Society of America, Research Triangle Park (1995)
10. 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 Cybernetics, Guangzhou, August 18-21 (2005)*
11. Caputo, M.: Linear model of dissipation whose Q is almost frequency independent—II. *Geophysical Journal of the Royal Astronomical Society* 13, 529–539 (1967)
12. Caputo, M.: *Elasticitae Dissipacione*. Zanichelli, Bologna (1969)
13. 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)*
14. Lubich, C.H.: Discretized fractional calculus. *SIAM Journal on Mathematical Analysis* 17(3), 704–719 (1986)
15. Mozaffari, A., Gorji-Bandpy, M., Gorji, T.B.: Optimal design of constraint engineering systems: application of mutable smart bee algorithm. *International Journal of Bio-inspired Computation* 4(3), 167–180 (2012)
16. Panda, R., Dash, M.: Fractional generalized splines and signal processing. *Signal Processing* 86(9), 2340–2350 (2006)

Artificial Bee Colony-Based Approach for Optimal Capacitor Placement in Distribution Networks

Attia El-Fergany¹, Almoataz Y. Abdelaziz², and Bijaya Ketan Panigrahi³

¹ Department of Electrical Power & Machines, Faculty of Engineering,
Zagazig University, 44519 Zagazig, Egypt

² Department of Electrical Power & Machines, Faculty of Engineering,
Ain Shams University, Cairo, Egypt

³ Department of Electrical Engineering, Indian Institute of Technology,
110016 New Delhi, India

Abstract. This manuscript presents an approach to allocate static capacitors along radial distribution networks using the artificial bee colony algorithm. In general practice the high potential buses for capacitor placement are initially identified using loss sensitivity factors. However, that method has proven less than satisfactory as loss sensitivity factors may not always indicate the appropriate placement. In the proposed approach, the algorithm identifies optimal sizing and placement and takes the final decision for optimum location within the number of buses nominated. The result is enhancement of the overall system stability index and potential achievement of maximum net savings. The obtained results are compared with those achieved using recent heuristic methods and show that the proposed approach is capable of producing high-quality solutions.

1 Introduction

Reactive power addition can be beneficial only when correctly applied. Correct application means choosing the correct position and size of the reactive power support. It is unmanageable to achieve zero losses in a power system, but it is likely to keep them to a minimum rate [1-3] in order to reduce the system overall costs. Capacitive reactive power support is used for the reduction of power losses together with other benefits; such as increased utilization of equipment, unloading of overloaded system components, and stopping the premature aging of the equipment.

A comprehensive survey of the literature from the last decade focusing on the various heuristic optimization techniques applied to determine the OCP and size is presented in [4]. Several heuristic tools that facilitate solving capacitor allocation optimization problems that were previously difficult or impossible to solve have been developed in the last decade [5-18].

Algorithms for enhancing voltage stability of electrical systems by OCP have been developed [19, 20], a relationship between voltage stability and loss minimisation was developed and the concept of maximising voltage stability through loss minimisation was outlined [21, 22].

In this paper, an ABC-based approach is utilized to ascertain the optimal size and select optimum locations of shunt capacitors. High potential buses for capacitor placement are initially identified by the observations of LSF with weak VSI of buses. The proposed method improves the voltage profile and reduces system losses in addition to enhancing voltage stability. The method has been tested and validated on a variety of radial distribution systems and the detailed results are presented. Different simplified methods of normal load distribution flow and other special techniques have been proposed [23, 24]. The distribution power flow suggested in [24] is used in this study.

2 Static Voltage Stability Index

Many different indices have been introduced to evaluate the power systems security level from the point of voltage static stability [25-28]. A new steady state VSI is proposed [28] for identifying the node, which is most sensitive to voltage collapse and expressed in Eq. (1) is utilised in the work. Fig. 1 shows the simple electrical equivalent of the radial distribution system.

$$VSI(j) = |V_i|^4 - 4[P_j \cdot X_{ij} - Q_j \cdot R_{ij}]^2 - 4[P_j \cdot R_{ij} + Q_j \cdot X_{ij}] \cdot |V_i|^2 \tag{1}$$

For stable the operation of the radial distribution networks, $VSI(j) \geq 0$. The node, at which the value of the VSI has lower value, is more sensitive to collapse. The node with the smallest VSI is the weakest node and the voltage collapse phenomenon will start from that node. Therefore, to avoid the possibilities of voltage collapse; the VSI of nodes should be maximized.

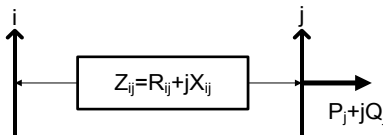


Fig. 1. line i-j power system model

3 Modeling of Objective Function and Constraints

The objective of capacitor allocations in the distribution system is to maximise the active power loss reduction, maximise the revenue of capacitor allocations and to enhance the system static stability subject to specific operating constraints. The objective function is mathematically formulated as,

$$Maximise \left\{ \begin{aligned} & w \cdot \left\{ C_e \cdot (P_{Lb} - P_{La}) \cdot T - \alpha \times \left[C_{Ci} \cdot N_B + C_C \cdot \sum_{i=1}^{N_B} Q_C(i) \right] \right\} \\ & + (1 - w) \cdot \left\{ \mu_F \cdot \sum_{j=2}^N VSI(j) \right\} \end{aligned} \right\} \tag{2}$$

Subject to the satisfaction of the Active and reactive power flow balance equations and a set of inequality constraints;

3.1 Power Balance Constraints

Power balance (Active and Reactive) constraints, which are equality constraints and include two nonlinear recursive power flow equations, can be formulated as follows,

$$\left. \begin{aligned} P_{Slack} &= \sum_{i=1}^{n_l} P_D(i) + \sum_{j=1}^n P_L(j) \\ Q_{Slack} + \sum_{i=1}^{N_B} Q_C(i) &= \sum_{i=1}^{n_l} Q_D(i) + \sum_{j=1}^n Q_L(j) \end{aligned} \right\} \quad (3)$$

3.2 Voltage Limit Constraint

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

$$V_{i,min} \leq |V_i| \leq V_{i,max}, i = 1 \dots N \quad (4)$$

3.3 Reactive Compensation Limit

Reactive power constraint in which injected reactive power at each candidate bus must be within their permissible ranges.

$$Q_{Ci}^{min} \leq Q_{Ci} \leq Q_{Ci}^{max}, i = 1 \dots N_B \quad (5)$$

3.4 Line Capacity Limit

The apparent power flow through the line S_l is restricted by its maximum rating limit as,

$$S_{li} \leq S_{li}^{rated}, \quad i = 1 \dots n \quad (6)$$

3.5 Maximum Total Compensation

From practical limitation, maximum compensation by using capacitor bank is limited to the total load reactive power demand.

$$\sum_{i=1}^{N_B} Q_C(i) \leq \sum_{j=1}^{n_l} Q_D(j) \quad (7)$$

3.6 Overall System Power Factor

System Power Factor should be maintained within desirable lower and upper limits.

$$PF_{min} \leq PF_{overall} \leq PF_{max} \quad (8)$$

A penalty factor associated with each violated constraint is added to the objective function in order to force the solution to stay away from the infeasible solution space; to respect the inequality constraints. Therefore, the optimal solution is established when no constraints is violated or even with acceptable tolerance and the objective function is maximised.

4 Identification of Potential Buses Using LSF

The estimation of these candidate nodes basically helps in significant reductions of the search space for the optimization procedure. In this proposed work, LSF is utilised for this purpose [22].

The LSF may be able to predict which bus will have the greatest loss reduction when reactive compensation is put in place. Consider a distribution line connected between 'i' and 'j' buses as shown in Fig. 1.

Active power loss in the ij^{th} line between i-j buses is given as shown in (9),

$$P_{ij-loss} = \frac{(P_j^2 + Q_j^2)}{|V_j|^2} \cdot R_{ij} \quad (9)$$

Thus, the sensitivity analysis factor is a derivative of the power loss with reactive power Q_j , as indicated in (10),

$$\frac{\partial P_{ij-loss}}{\partial Q_j} = \frac{2 \times Q_j}{|V_j|^2} \cdot R_{ij} \quad (10)$$

The values are arranged in descending order for all the lines of the given system. The descending order of the elements vector will decide the sequence in which the buses are to be considered for compensation. Buses of higher LSF and lower VSI have a greater chance of being selected as candidate locations for capacitor installations.

5 Artificial Bees Colony Algorithm

The ABC algorithm was proposed by Karaboga for optimizing numerical complex problems [29]. It simulates the intelligent foraging behaviour of honey bee swarms. It is a very simple, robust and population based stochastic optimization algorithm in nature. The performance of the ABC algorithm has been compared with those of other well-known modern heuristic algorithms such as GA, DE and PSO on constrained and unconstrained problems [30, 31]. The algorithm has a well-balanced exploration and exploitation ability. Recent enhancements of ABC have been proposed [32-34] to improve performance of sharing information between artificial bees and to enhance its performance.

A bee carrying out random search is called scout. In the ABC algorithm, first half of the artificial colony consists of employed bees and the second half constitutes the artificial onlookers. For every food source, there is only one employed artificial bee. The employed bee whose food source is exhausted by the employed and onlooker bees becomes a scout.

At initialization stage, a set of food source positions are randomly selected by the artificial bees and their nectar amounts are determined. These bees come into hive and share the nectar information of sources with the bees waiting on the dance area within the hive.

After sharing the information, every employed bee goes to the food source area visited by her at the previous cycle since that food source exists in her memory, and

then chooses a new food source by means of visual information in the neighborhood of the present one. Then an onlooker prefers a food source area depending on the nectar information distributed by the employed bees on the dance area. As the nectar amount of a food source increases, the chance with which that food source is chosen by an onlooker increases, too.

After arriving at the selected area, employed bee chooses a new food source in the neighborhood of the one in the memory depending on visual information. Visual intelligent evidence is based on the comparison of food source positions. When the nectar of a food source is abandoned by the bees, a new food source is randomly determined by a scout bee and replaced with the abandoned one. In this model, at each cycle one scout goes outside for searching a new food source and the number of employed and onlooker bees were equal. The probability P_i of selecting a food source i is determined by using

$$P_i = \frac{fit_i}{\sum_{n=1}^{S_N} fit_n} \quad (11)$$

After all artificial onlookers have selected their food sources; each of them determines a food source in the neighborhood of his chosen food source and computes its fitness.

The best food source among all the neighboring food sources determined by the onlookers associated with a particular food source i will be the new location of the food source i . If a solution represented by a particular food source does not improve for a predetermined number of iterations then that food source is abandoned by its associated employed bee and it becomes a scout. This tantamount to assigning a randomly generated food source to this scout and changing its status again from scout to employed. After the new location of each food source is determined, another iteration of ABC algorithm begins.

The whole process is repeated again and again till the termination condition is satisfied. The food source in the neighborhood of a particular food source is determined by altering the value of one randomly chosen solution parameter and keeping other parameters unchanged.

The procedure of the ABC algorithm to solve OCP can be summarized in the flow chart diagram of Fig. 2.

6 Numerical Results and Simulations

In order to test the effectiveness and performance of the proposed ABC-based algorithm, it has been applied to several distribution radial test systems. Due limitations of space supposed by submission guidelines, only the 34-bus radial distribution system is selected for reporting and demonstration in this article, to test and examine the applicability of the proposed approach. Simulations are carried out using MATLAB environment, release of 2011a/®7.12 and executed on a Laptop with Processor Intel® Core i5 CPU 2.40 GHz with a 4.0 GB of RAM with 32-bit operating system. In all calculations; for all the test cases, the following constants are assumed and applied as shown in Table 1.

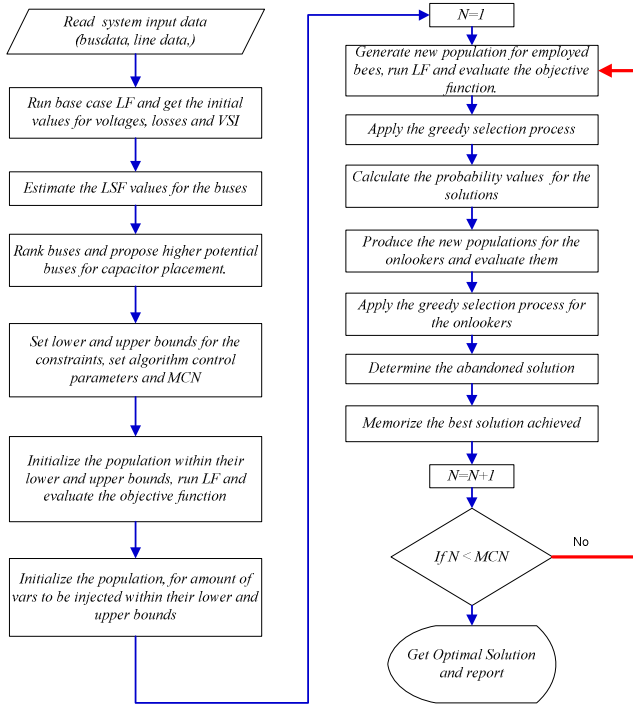


Fig. 2. Flow Chart of ABC Algorithm and capacitor allocations

Table 1. Constants for the rates using a long with test cases

| SN | Item | Proposed rate |
|----|---------------------|---------------------|
| 1 | Average energy cost | \$0.06/kWh |
| 2 | Depreciation factor | 20% |
| 3 | Purchase cost | \$25/kVAr |
| 4 | Installation cost | \$1,600/location |
| 5 | Operating cost | \$300/year/location |
| 6 | Hours per year | 8760 |

The net savings are calculated using:

$$\text{Net} \frac{\text{Savings}}{\text{year}} = \left\{ \alpha \times \{ \text{Cost of Installations} + \text{Cost of Purchase} \} - \frac{\text{Cost of Energy Reductions} - \text{Operating cost/year}}{\text{year}} \right\} \quad (12)$$

It is well-known that LSF observations, may not lead to the optimum locations. Due to the fact the LSF calculations depend on the network topology, configurations, loading, etc... and to tackle these limitations, the algorithm will search the optimum number of buses and select them for capacitor placements [35]. The proposed method has been programmed and implemented using MATLAB [36, 37].

34-Bus Test System: Numerical Results and Simulations

This 34-bus test case has 4-lateral radial distribution system which is shown in Fig. 3. The data of the system are obtained from [2]. The total load of the system is (4,636.5+j2,873.5) kVA.

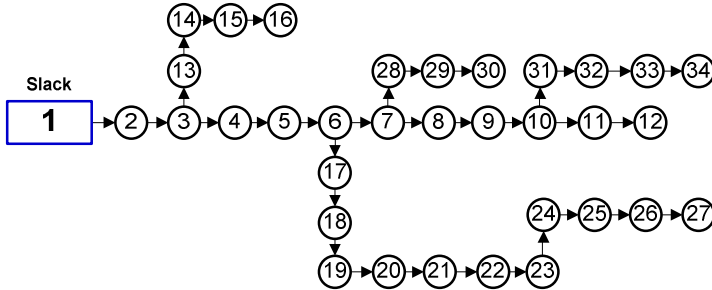


Fig. 3. Single line diagram of a 34-bus radial distribution network

Using base LF to candidate the potential buses for capacitor placement and based on LSF and VSI values; as follows, [19, 22, 20, 21, 23, 24, 25, 26 & 27]. Parameters adopted for the ABC algorithm for the test case of a 34-bus, and the required inequality constraints to be relaxed and respected are given in Table 2.

Table 2. Control parameters adopted for the ABC algorithm and target setting for the constraints

| Item | Proposed Setting/bounds |
|---------------------------|--|
| Swarm size (SN) | 60 |
| Limit | 30 |
| MCN | 100 |
| Bus Voltage constraint | $0.95 \leq v_i \leq 1.05$ |
| Power Factor Constraint | $0.95 \leq PF_{overall} \leq 0.99$ |
| Allowable capacitor range | 0 kVAr to 1500 kVAr with step of 50 kVAr |

After running the proposed optimization algorithm to select the optimal locations and determine the capacitor optimal sizes, the outcome leads to only 2 locations for capacitor placement which are buses 19 and 24 with optimum capacitor ratings of 950 kVAr and 900 kVAr, respectively. The CPU computational time needed is 10.08 s to accomplish this optimization process by the proposed ABC-based method. The results of the proposed method compared with the results of GA [12], PSO [6], HS-based [2], PGSA [15] and EA [38] for the reactive compensation required and relevant bus allocations are shown in Table 3.

For comparison purposes, the reported figures in [12], [6], [2], [15] and [38] of reactive power at specific buses are recycled to calculate the system losses and the net savings (refer to Table 4) with the same rates proposed in this article as shown in Table 1 and Eq. (12).

The system overall power factor is significantly corrected from 0.856 lagging (base case) to 0.980 lagging with capacitor allocations, respectively. The VSI of a 34-bus radial distribution system without and with compensations is depicted Fig. 4.

Table 3. Optimal Location of Capacitor placement and value of capacitor size

| Method | ABC | GA [12] | PSO [6] | HS [2] | PGSA [15] | EA [38] |
|--------------------------|-----------|-----------|-----------|------------|------------|-----------|
| (Location, Size in kVAr) | (19, 950) | (5, 300) | (19, 781) | (26, 1400) | (19, 1200) | (8, 1050) |
| | (24, 900) | (9, 300) | (22, 803) | (11, 750) | (22, 639) | (18, 750) |
| | | (12, 300) | (20, 479) | (17, 300) | (20, 200) | (25, 750) |
| | | (22, 600) | | (4, 250) | | |
| | | (26, 300) | | | | |

Table 4. Results and comparisons of a 34-Bus radial feeder test case ($w = 0.5$)

| Item | Without OCP | With OCP | | | | | |
|----------------------------|-------------|----------|----------|----------|----------|-----------|----------|
| | | ABC | GA [12] | PSO [6] | HS [2] | PGSA [15] | EA [38] |
| VSI_{min}^a | 0.7860 | 0.8130 | 0.8071 | 0.8097 | 0.8219 | 0.8074 | 0.8149 |
| VSI_{max}^a | 0.9765 | 0.9797 | 0.9796 | 0.9800 | 0.9811 | 0.9800 | 0.9808 |
| $\sum_{j=2}^{34} VSI(j)$ | 28.621 | 30.122 | 29.089 | 29.135 | 29.321 | 29.115 | 29.268 |
| P_{loss} (kW) | 221.74 | 167.99 | 164.96 | 169.36 | 168.48 | 171.96 | 161.27 |
| Reductions in P_{loss} % | --- | 24.24% | 25.61% | 23.62% | 24.02% | 22.45% | 27.27% |
| PF _{overall} | 0.856 | 0.980 | 0.983 | 0.997 | 0.999 | 0.974 | 0.984 |
| $\sum Q_c$ (kVAr) | --- | 1,850 | 1,800 | 2,063 | 2,700 | 2,039 | 2,550 |
| Net Savings/year | --- | \$17,756 | \$15,093 | \$15,570 | \$12,017 | \$15,590 | \$17,173 |

^aExcluding slack bus # 1

From the results illustrated and shown in table 4, the ABC algorithm yields to reduce peak losses to 167.99 kW with 1,850 kVAr installed at 2 locations only (buses 19 & 24), which are optimally selected using the current methodology out of 9 initial higher potential buses estimated by LSF calculations. The net savings gained are **\$15,093, \$15,570, \$12,017, \$15,590, \$17,173 and \$17,756** using GA [12], PSO [6], HS [2], PGSA [15], EA [38] and the ABC algorithm, respectively.

If the objective is being to minimise the active power loss only (high potential buses are pre-identifies as outlined in section 4 using LSF) or to maximise VSI only (high potential buses are of those with lower VSI), irrespective the cost savings, while maintaining the equality and inequality constraints. Table 5 depicts the extracted summaries for the cases of VSI maximization and P_{loss} minimization as well. In the case of VSI maximization, the nominated buses for capacitor allocations are identified based on lowest buses VSI values.

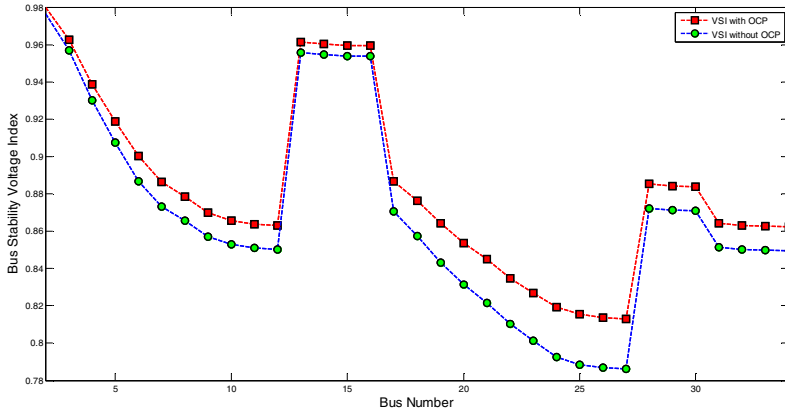


Fig. 4. VSI values of buses against bus number with and without OCP (2 locations)

Table 5. Summaries of the VSI and P_{loss} objectives

| Item | Power loss Minimisation ($w = 1$ & $\alpha = 0$) | VSI Maximisation ($w = 0$) |
|--------------------------|---|---------------------------------|
| VSI_{min}^a | 0.812 | 0.824 |
| VSI_{max}^a | 0.981 | 0.981 |
| $\sum_{j=2}^{34} VSI(j)$ | 29.2815 | 30.3061 |
| P_{loss} (kW) | 161.087 | 169.92 |
| $PF_{overall}$ | 0.9978 | 0.9952 |
| $\sum Q_c$ (kVAr) | 2,600 (3 locations) | 2,450 (8 locations) |
| <i>Net Savings/year</i> | \$17,018 | \$8,785 |

^aExcluding slack bus # 1

7 Conclusion

The application of the ABC optimization approach for solving the problem of capacitor allocations (sizing and location) to maximise the net benefits and to improve system static voltage stability has been presented and investigated. The numerical results of the simulation indicate a considerable improvement in active power losses reductions, voltage stability enhancements, and power factor corrections while maximizing the net savings. The achieved results via the ABC-based method are competitive compared to the other recent heuristic methods in terms of the quality of the solution and computational efficacy. The main advantage of the ABC algorithm is that it does not require expending more effort in tuning the control parameters, as in the case of GA, DE, PSO and other EAs. This feature marks the ABC-based algorithm as being advantageous for implementation.

References

- [1] Haque, M.H.: Capacitor placement in radial distribution systems for loss reduction. *IEE Proc. Gener. Transm. Dis.* 146(5), 501–505 (1999)
- [2] Chis, M., Salama, M., Jayaram, S.: Capacitor placement in distribution system using heuristic search strategies. *IEE Proc. Gener. Transm. Dist.* 144(3), 225–230 (1997)
- [3] Short, T.A.: *Electric power distribution equipment and systems capacitor application.* Taylor & Francis Group (2005)
- [4] Sirjani, R., Azah, M., Shareef, H.: Heuristic optimization techniques to determine optimal capacitor placement and sizing in radial distribution networks: a comprehensive review. *Przegląd Elektrotechniczny (Electr. Rev.)* 88(7a), 1–7 (2012)
- [5] Gallego, R.A., Monticelli, A.J., Romero, R.: Optimal capacitor placement in radial distribution networks using tabu search. *IEEE Trans. Power Syst.* 16(4), 630–637 (2001)
- [6] Prakash, K., Sydulu, M.: Particle swarm optimization based capacitor placement on radial distribution systems. *IEEE PES General Meeting*, 1–5 (2007)
- [7] Yu, X.M., Xiong, X.Y., Wu, Y.W.: A PSO-based approach to optimal capacitor placement with harmonic distortion consideration. *Electr. Power Syst. Res.* 71, 27–33 (2004)
- [8] Sirjani, R., Mohamed, A., Shareef, H.: Optimal capacitor placement in a radial distribution system using harmony search algorithm. *J. Appl. Sci.* 10(23), 2996–3006 (2010)
- [9] 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)
- [10] Annaluru, R., Das, S., Pahwa, A.: Multi-level ant colony algorithm for optimal placement of capacitors in distribution systems. In: *Congress on Evol. Comput., CEC*, vol. 2, pp. 1932–1937 (2004)
- [11] Chiang, H.D., Wang, J.C., Cockings, O., Shin, H.D.: Optimal capacitor placements in distribution systems: part 1: a new formulation and the overall problem. *IEEE Trans. Power Deliv.* 5(2), 634–642 (1990)
- [12] Swarup, K.S.: Genetic Algorithm for optimal capacitor allocation in radial distribution systems. In: *Proc. the 6th WSEAS Int. Conf. Evolut. Comput., Lisbon, Portugal, June 16–18*, pp. 152–159 (2005)
- [13] Hsiao, Y.T., Chen, C.H., Chien, C.C.: Optimal capacitor placement in distribution systems using a combination fuzzy-GA method. *Electr. Power Energy Syst.* 26, 501–508 (2004)
- [14] Tabatabaei, S.M., Vahidi, B.: Bacterial foraging solution based fuzzy logic decision for optimal capacitor allocation in radial distribution system. *Electr. Power Syst. Res.* 81, 1045–1050 (2011)
- [15] Rao, R.S., Narasimham, S.V.L., Ramalingaraju, M.: Optimal capacitor placement in a radial distribution system using plant growth simulation algorithm. *Int. J. Electr. Power Energy Syst.* 33(5), 1133–1139 (2011)
- [16] Huang, S.: An immune-based optimization method to capacitor placement in a radial distribution system. *IEEE Trans. Power Deliv.* 15(2), 744–749 (2000)
- [17] El-Fergany, A.: Optimal capacitor allocations using integrated evolutionary algorithms. *IET Gener. Transm. Distrib.* 7(6), 593–601 (2013)
- [18] Sedighzadeh, M., Arzaghi-haris, D.: Optimal allocation and sizing of capacitors to minimize the distribution line loss and to improve the voltage profile using big bang-big crunch optimization. *Int. Rev. Electr. Eng. (IREE)* 6(4), Part B, 2013–2019 (2011)
- [19] El Arini, M.: Optimal capacitor placement incorporating voltage stability and system security. *Eur. Trans. Electr. Power* 10(5), 319–325 (2000)

- [20] Satpathy, P.K., Das, D., Gupta, P.B.: Critical switching of capacitors to prevent voltage collapse. *Electr. Power Sys. Res.*, 11–20 (2004)
- [21] Jasmon, B., Lee, L.H.C.C.: Maximising voltage stability in distribution networks via loss minimization. *J. Electr. Power Energy Sys.* 13(3), 148–152 (1991)
- [22] Graham, W.A., McDonald, J.R.: Planning for distributed generation within distribution networks in restructured electricity markets. *IEEE Power Eng. Rev.*, 52–54 (2000)
- [23] Teng, J.H.: A direct approach for distribution system load flow solutions. *IEEE Trans. Power Deliv.* 18(3), 882–887 (2003)
- [24] Gözel, T., Eminoglu, U., Hocaoglu, M.H.: A tool for voltage stability and optimization in radial distribution systems using MATLAB GUI. *Simul. Model Pract. Theory* 16(5), 505–518 (2008)
- [25] Moghavammi, M., Omar, F.M.: Technique for contingency monitoring and voltage collapse prediction. *IEE Proc. Gener. Transm. Dist.* 145, 634–640 (1998)
- [26] Musirin, I., Rahman, T.K.A.: Estimating maximum loadability for weak bus identification using FVSI. *IEEE Power Eng. Rev.* 22, 50–52 (2002)
- [27] Anhit, S., Ndarajah, M., Kwang, S.: A maximum loading margin method for static voltage stability in power systems. *IEEE Trans. Power Syst.* 21(2), 965–972 (2006)
- [28] Charkravorty, M., Das, D.: Voltage stability analysis of radial distribution networks. *Int. J. Electr. Power Energy Syst.* 23(2), 129–135 (2001)
- [29] 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)
- [30] Karaboga, D., Basturk, B.: On the performance of artificial bee colony algorithm. *Appl. Soft Comput.* 8, 687–697 (2008)
- [31] Karaboga, D., Akay, B.: A comparative study of Artificial Bee Colony algorithm. *J. Appl. Math. Comput.* 214, 108–132 (2009)
- [32] Das, S., Biswas, S., Kundu, S.: Synergizing fitness learning with proximity-based food source selection in artificial bee colony algorithm for numerical optimization. *Appl. Soft Comput.* (in press, 2013), doi:10.1016/j.asoc.2013.07.009
- [33] Biswas, S., Kundu, S., Das, S., Vasilakos, A.: Information sharing in bee colony for detecting multiple niches in non-stationary environments. In: *GECCO 2013 Companion Proceeding of the Fifteenth Annual Conference Companion on Genetic and Evolutionary Computation Conference Companion* (2013), doi:10.1145/2464576.2464588
- [34] Mozaffari, A., Gorji-Bandpy, M., Gorji, T.B.: Optimal design of constraint engineering systems: application of mutable smart bee algorithm. *International Journal of Bio-inspired Computation* 4(3), 167–180 (2012)
- [35] El-Fergany, A., Abdelaziz, A.Y.: Capacitor placement for net saving maximization and system stability enhancement in distribution networks using artificial bee colony-based approach. *Int. J. Electr. Power Energy Syst.* 54, 235–243 (2014)
- [36] <http://www.mathworks.com>
- [37] <http://mf.erciyes.edu.tr/abc/>
- [38] Ali, E., Boudour, M., Rabah, G.: New evolutionary technique for optimization shunt capacitors in distribution networks. *J. Electr. Eng.* 62(3), 163–167 (2011)

Nomenclatures

| | |
|------------------|--|
| N | total network buses |
| P_{Loss} | total network active loss |
| $VSI(j)$ | voltage stability index of bus j |
| R_{ij} | resistance of line i-j |
| X_{ij} | reactance of line i-j |
| $ V_i $ | voltage magnitude of bus i |
| $ V_j $ | voltage magnitude of bus j |
| P_j | total effective real power load fed through bus j |
| Q_j | total effective reactive power fed through bus j |
| C_e | energy cost |
| T | time period |
| P_{La} | total active power loss after compensation |
| P_{Lb} | total active peak power loss before compensation |
| C_{Ci} | cost of installation |
| N_B | number of candidate effective buses (that have compensations > 0) |
| C_C | cost of the capacitor |
| α | depreciation factor |
| μ_F | magnifying factor |
| n | number of lines |
| n_l | number of load buses |
| W | weighting factor |
| P_{Slack} | active power supplied from the slack bus |
| Q_{Slack} | reactive power supplied from the slack bus |
| $P_D(i)$ | active power demand of load at bus i |
| $Q_D(i)$ | reactive power demand of load at bus i |
| $P_L(j)$ | active power loss at branch j |
| $Q_L(j)$ | reactive power loss at branch j |
| $Q_C(i)$ | amount of reactive power of installed capacitors at bus i |
| $V_{i,min}$ | lower permissible voltage limit at bus i |
| $V_{i,max}$ | upper permissible voltage limit at bus i |
| Q_{Ci}^{min} | lower reactive power limit of compensated bus i |
| Q_{Ci}^{max} | upper reactive power limit of compensated bus i |
| S_{li} | actual line flow of line i |
| S_{li}^{rated} | rated line transfer capacity |
| PF_{min} | lower limit of overall system power factor at substation (slack bus) |
| PF_{max} | upper limit of overall system power factor at substation (slack bus) |
| fit_i | fitness of the solution represented by food source i |
| SN | total number of food sources |
| MCN | maximum cycle number |

List of Abbreviations

| | |
|-------------|-----------------------------------|
| <i>ABC</i> | artificial bees colony |
| <i>DE</i> | differential evolution |
| <i>GA</i> | genetic algorithm |
| <i>LSF</i> | loss sensitivity factor |
| <i>OCP</i> | optimal capacitor placement |
| <i>PGSA</i> | plant growth simulation algorithm |
| <i>PSO</i> | particle swarm optimization |
| <i>PS</i> | pattern search |
| <i>VSI</i> | voltage stability index |
| <i>HS</i> | heuristic search |
| <i>EA</i> | evolutionary algorithm |
| <i>P.U.</i> | per unit |
| <i>LF</i> | load flow |

Grammatical Bee Colony

Tapas Si^{1,*}, Arunava De², and Anup Kumar Bhattacharjee³

¹ Department of Computer Science & Engineering
Bankura Unnayani Institute of Engineering, Bankura, West Bengal, India
c2.tapas@gmail.com

² Department of Information Technology

Dr. B.C Roy Engineering College, Durgapur, W.B, India

³ Department of Electronics and Communication Engineering
National Institute of Technology, Durgapur, West Bengal, India

Abstract. This paper presents Grammatical Bee Colony algorithm. Grammatical Bee Colony is variant of Grammatical Evolution algorithm in which Artificial Bee Colony is used as search engine to write a program in any arbitrary language. The performance of Grammatical Bee Colony is tested on benchmark problems. Experimental results shows that Grammatical Bee Colony is able to generate programs.

Keywords: Artificial bee colony, Grammatical evolution, Grammatical bee colony, Grammatical differential evolution, Grammatical swarm.

1 Introduction

Grammatical Evolution (GE) [1,2] is developed by C. Ryan *et al.* in 1998. GE is a form of Genetic Programming (GP) [3] which can write programs in any arbitrary language. Variable-length genetic algorithm is used in genotype-to-phenotype mapping in standard GE. O'Neill *et al.* proposed Grammatical Swarm (GS)[5] algorithm. Particle swarm optimization(PSO) [6] algorithm is used as a learning algorithm in GS to generate the programs in any arbitrary language. O'Neill *et al.*[7] proposed Grammatical Differential Evolution(GDE) which is another variant of GE algorithm. GDE algorithm adopts Differential Evolution(DE) learning algorithm.

The objective of this paper is to use Artificial Bee Colony [8] algorithm as a search engine to generate computer programs. The devised method is termed as Grammatical Bee Colony (GBC) algorithm in this paper. GBC algorithm is tested to solve benchmark problems taken from Genetic Programming literature.

Organization of this Paper. Grammatical Evolution along with its different learning algorithms is described in Section 2. Grammatical Bee Colony is discussed in Section 3. Experiments and Results are described in Section 4. Finally a conclusion is given in Section 5.

* Corresponding author.

2 Grammatical Evolution and Its Learning Algorithms

Genetic Programming(GP) was invented by Koza [3] in the year 1992 to evolve the computer programs automatically. GP is a form of Genetic Algorithm(GA) in which each genome is a tree structure used to represent computer program. GE is a form of grammar based genetic programming [4] in which Context Free Grammar(CFG) is combined with GP to put restrictions in search space according to problems of interest and derivation trees are constructed from the predefined CFG. Genomes are variable-length linear array in forms of binary strings in GE. Bankus-Naur Form (BNF) of CFG is used to map genotype-to-phenotype for generating programs in any arbitrary language. There are two variants of Grammatical Evolution with respect to learning algorithm and they are Grammatical Swarm and Grammatical Differential Evolution.

Grammatical Swarm (GS) is a variant of Grammatical Evolution. In GS, particle swarm optimization (PSO) [6] algorithm is used as learning algorithm for generating computer programs through *genotype-to-phenotype mapping* process. Each particle in particle swarm optimizer represents a set of integer codons from which computer programs are generated using BNF grammatical rules.

Grammatical Differential Evolution(GDE) is a GS like variant of Grammatical Evolution algorithm. In GDE, Differential Evolution (DE) algorithm is used as search engine in generating computer programs. Each vector in DE represents a set of integer codons to map from genotype to phenotype using BNF grammatical rules.

The both GDE and GS became unable to outperform over GE algorithm. This paper makes an attempt to use Artificial Bee Colony (ABC) algorithm as a learning algorithm in GE. The description of ABC algorithm is given in the next section.

3 Grammatical Bee Colony

Artificial Bee Colony(ABC) [8] is successfully used in Artificial Bee Colony Programming(ABCP) proposed by D. Karaboga *et al.* [9]. In ABCP, ABC algorithm is used in program induction for symbolic regression. Food source's position in ABC represents computer programs by trees as like in GP. Grammatical Bee Colony (GBC) uses Artificial Bee Colony (ABC) algorithm as a search engine to generate computer programs through genotype-to-phenotype mapping using BNF of Context-Free Grammar. A solution in ABC algorithm is represented as food source's position. Obviously, a food source's position represents a genotype in GBC. Using BNF of Context-Free Grammar, programs (phenotype) are generated from the food source's position.

3.1 Artificial Bee Colony

D. Karaboga [8] developed Artificial Bee Colony(ABC) algorithm which is an optimization algorithm based on the foraging behaviour of honey bees. In this

algorithm, food source represents a solution of a problem. Artificial bee colony has three groups of bees and they are employed bees, onlookers and scout. Employed bees search new food source in their neighbourhood of food source in their memory. Onlookers take information from employed bees in the hive and select one of the food sources. They search new food by themselves in the neighbourhood of food source. An employed bee become a scout when its food source is abandoned and again searches for new food source. “limit” is one important control parameter in ABC algorithm and it is used to determine, after how many trials, employed bees will become scouts. A detail description of ABC algorithm is given in Table 1.

Table 1. Artificial Bee Colony Algorithm

-
1. Initialize a population $X_i (i = 1, 2, \dots, n)$
 2. Evaluate fitness F_i
 3. **while** ($t < CYCLE_{max}$) or (stop criterion)
 4. Generate new solutions (food source position) V_{ij} in the neighbourhood of X_{ij} for the employed bees using the following equation:

$$V_{ij} = X_{ij} + \Phi_{ij}(X_{ij} - X_{kj}) \quad (1)$$

where k is the neighbourhood of i , Φ_{ij} is a random number in the range $[-1, 1]$ and evaluate them.

5. Select better solution from X_{ij} and V_{ij}
6. Calculate the probability values for P_i for the solutions X_{ij} by means of their fitness values using the following equation:

$$P_i = \frac{fit_i}{\sum_{i=1}^{NP} fit_i} \quad (2)$$

fitness is calculated by the following equation:

$$fit_i = \begin{cases} \frac{1}{1+F_i} & \text{if } F_i \geq 0 \\ 1 + |F_i| & \text{if } F_i < 0 \end{cases} \quad (3)$$

9. Normalize the P_i values in the range (0,1)
8. Generate new solutions V_{ij} for the onlookers from the solution X_{ij} selected depending on P_i and evaluate them
9. Select better solution from X_{ij} and V_{ij}
10. Determine the abandon solution after “limit” and replace that by a new randomly generated solution for the scouts using the following equation:

$$X_{ij} = X_{min} + (X_{max} - X_{min}) \times rand(0, 1) \quad (4)$$

11. Evaluate new solutions and find the best solution so far
 12. **end while**
-

3.2 Backus-Naur Form of Context-Free Grammar

The Backus-Naur Form (BNF) of Context-Free Grammar(CFG) is used for genotype-phenotype mapping. BNF is a metasyntax used to express Context-Free Grammar. An example of BNF of CFG is described below:

1. $\langle \text{expr} \rangle := (\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle)$ (0)
 | $\langle \text{var} \rangle$ (1)
2. $\langle \text{op} \rangle := +$ (0)
 | $-$ (1)
 | $*$ (2)
 | $/$ (3)
3. $\langle \text{var} \rangle := x_1$ (0)
 | x_2 (1)

3.3 Genotype and Phenotype

In Grammatical Bee Colony, food source's position is used to represent a genome and it is a set of integer codons in the range [0,255]. A part of genotype representation is given in Fig.1 for an example.

| | | | | | |
|------------|-----------|------------|-----------|------------|------------|
| 152 | 58 | 160 | 97 | 221 | 186 |
|------------|-----------|------------|-----------|------------|------------|

Fig. 1. Genotype Representation

A *mapping process* is used to map from integer-value to rule number in the derivation of expression using BNF of Context-Free Grammar by the following ways:

rule=(codon integer value) MOD (number of rules for the current non-terminal)

In the derivation process, if the current non-terminal is $\langle \text{expr} \rangle$, then, the rule number is generated by the following way:

rule number=(152 mod 2)=0

$\langle \text{expr} \rangle$ will be replaced by ($\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle$)

$\langle \text{expr} \rangle := (\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle)$ (152 mod 2)=0
 $:= (\langle \text{var} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle)$ (58 mod 2)=0
 $:= (x_1 \langle \text{op} \rangle \langle \text{expr} \rangle)$ (160 mod 4)=0
 $:= (x_1 + \langle \text{expr} \rangle)$ (97 mod 2)=1
 $:= (x_1 + \langle \text{var} \rangle)$ (221 mod 2)=1
 $:= (x_1 + x_2)$

When derivation run out of codons, then the derivation again starts from the beginning of the genome. This process is called as *wrapping*. When same rule

number is generated again and again for a variable(for an example,<expr> is replaced by (<expr><op><expr>)). Therefore it will take indefinite time and the wrapping process may become failure. Therefore, in this work, wrapping process is done for two times. After wrapping process, if there are non-terminals remain in the derived string, it is denoted as invalid and its fitness is assigned a very small value so that it can be replaced by better valid individual later.

4 Experiments and Results

4.1 Benchmark Problems

Santa Fe Ant Trail Problem The Santa Fe Ant Trail is a standard benchmark problem in GP research. The objective of this problem is to evolve a computer program by which an artificial ant can find all 89 pieces of food in a non-continuous trail in 600 time steps. The trail is located on a 32×32 toroidal grid. The ant can run left, right move one square forward and may look ahead one square in its front to determine whether that square contains any food. All actions, except of looking ahead of food, take one time step. The ant starts from top-left corner of the grid facing the first piece of food on the trail. The fitness function is calculated as the difference between the total number of pieces of foods before and after the runs:

$$fitness = 89 - F$$

where F is the total number of pieces of food eaten by ant at the end of the run. The BNF of CFG for this problem is given in below:

1. <code> := (<code><line>) | <line>
2. <line> := <condition> | <op>
3. <condition> := if(food_ahead()){<line>}else{<line>}
4. <op> := left(); | right(); | move();

The generated program using the above grammar is placed in a loop with following termination criteria:

1. *Time Steps* = 600 or
2. Ant eats all 89 pieces of foods placed in the grid.

Symbolic Regression Problem. The target function is $f(x) = x + x^2 + x^3 + x^4$ and 100 fitness cases are generated randomly in the range $[0,1]$. The fitness function is calculated as the sum of absolute error over 100 fitness cases. The BNF of CFG for this problem is given in bellow:

1. <expr> := (<expr><op><expr>) | <var>
2. <op> := + | - | * | /
3. <var> := x

'/' is protected division.

Even 3-Parity Problem. Even 3-parity problem has 8 fitness cases and the fitness function is calculated as the sum of absolute error over 8 fitness cases. The BNF of CFG for this problem is given in bellow:

1. $\langle \text{expr} \rangle := (\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle) \mid \langle \text{var} \rangle$
2. $\langle \text{op} \rangle := \text{or} \mid \text{and} \mid \text{nor} \mid \text{nand}$
3. $\langle \text{var} \rangle := x1 \mid x2 \mid x3$

3-Multiplexer. 3-input multiplexer problem has been tested with GBC. 3-input multiplexer problem has 8 fitness cases and the fitness function is calculated as the sum of absolute error over 8 fitness cases. The BNF of CFG for this problem is given in bellow:

1. $\langle \text{expr} \rangle := (\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle) \mid \langle \text{var} \rangle$
2. $\langle \text{op} \rangle := \text{or} \mid \text{and} \mid \text{nor} \mid \text{nand}$
3. $\langle \text{var} \rangle := x1 \mid x2 \mid x3$

4.2 Parameters Settings

The parameters of GBC are set as following: Population size = 100, Dimension = 100, Number of cycle = 300, Limit = 5.

The parameters of GS are set as following: Population size = 30, Dimension = 100, Number of iteration = 1000, $V_{max} = 0.5 \times 255$, $c_1 = c_2 = 1.49445$, $\{\omega_{max}, \omega_{min}\} = \{0.9, 0.4\}$

The parameters of GDE/rand/1/bin are set as following: Population size = 500, Dimension = 100, Number of iteration = 60, Scale Factor = 0.8
Cross-over rate = 0.8.

Each of GBC, GS and GDE algorithms is allowed to run for 30,000 number of function evaluations(FEs) in a single run.

4.3 PC Configuration

1. System: Windows 7
2. CPU: AMD FX -8150 Eight-Core
3. RAM: 16 GB
4. Software: Matlab 2010b

The experiments are carried out for each problem with 50 separate runs. Mean and standard deviation of best-run-errors of each problem for Grammatical Swarm, Grammatical Differential Evolution and Grammatical Bee Colony are given in Table 2¹. From this results it has been seen that Grammatical Swarm performed better than Grammatical Differential Evolution and Grammatical Bee Colony in Santa Fe Ant Trail Problem. There are no difference in performances of

¹ Results in bold-face indicates better

Grammatical Differential Evolution as well as Grammatical Bee Colony in Even 3-Parity problem where as a near solution to them is produced by Grammatical Swarm. Grammatical Bee Colony performed better than Grammatical Swarm in multiplexer problem. Grammatical Differential Evolution performed better in symbolic regression problem than Grammatical Swarm and Grammatical Bee Colony. A *t-test* has been carried out between GBC and the best outcomes of GS and GDE for statistical significance test with 95% confidence interval and degree of freedom =98. There is no significance difference between GDE and GBC for even-3 parity and symbolic regression problems. Grammatical Swarm statistically outperformed over GBC for Ant problem whereas GDE statistically outperformed over GBC for multiplexer problem. Therefore these results demonstrate the proof of concept that Grammatical Bee Colony can generate programs successfully to solve problems of interest. The convergence graphs of GBC for each problem are given in Figure 2.

Table 2. Mean and standard deviation of best-run-errors

| Problems | GS | | GDE/rand/1/bin | | GBC | |
|---------------------|-------------|----------------|----------------|---------------|-------------|---------------|
| | Mean | Std. Dev. | Mean | Std. Dev. | Mean | Std. Dev. |
| Santa Fe Ant Trail | 25.3 | 3.30275 | 28.86 | 6.1578 | 32.30 | 11.1268 |
| Symbolic Regression | 7.98 | 6.4296 | 6.93 | 8.3208 | 8.22 | 6.6231 |
| Even 3-Parity | 2.00 | 0.00 | 1.98 | 0.1414 | 1.98 | 0.3187 |
| 3-Multiplexer | 0.92 | 0.1414 | 0.48 | 0.5046 | 0.82 | 0.3880 |

A GBC generated program for Santa Fe Ant Trail problem and using that program, ant eats 64 pieces of total 89 pieces of food

```

move();
if(foodahead()) right();
else if(foodahead()) left();
else if(foodahead())
  if(foodahead())
if(foodahead())
  if(foodahead()) right();
else left(); end;
  else right(); end;
  else if(foodahead()) right();
else if(foodahead()) left();
else if(foodahead())
if(foodahead()) left();
  else move(); end;
else move(); end;
end; end; end;

```

```

else left(); end; end; end;
if(foodahead()) if(foodahead()) move();
else if(foodahead())
if(foodahead()) if(foodahead())
if(foodahead()) if(foodahead())
if(foodahead()) right(); else right(); end;
  else if(foodahead()) right();
else if(foodahead()) if(foodahead())
if(foodahead()) move(); else right(); end;
else right(); end;
else if(foodahead()) right();
  else move(); end; end; end; end;
else move(); end; else right(); end;
else move(); end; else move(); end; end;
else right(); end;

```

*A GBC generated program for Symbolic Regression problem (absolute rror=
3.011480e-015)*

```

plus(pdivide(plus(times(x,times(times(x,x),x)),x),
pdivide(x,x)),times(plus(x,times(x,x)),x))

```

pdivide function is defined as follows to avoid division by zero error:

```

function value =pdivide(arg1,arg2)
if abs(arg2)<= 0.00001 value=arg1;
else value=arg1/arg2;
end
end

```

A GBC generated program for Even 3-Parity problem(absolute error=1)

```

nor(nor(x2, and(x3, x1)), and( and( and(x2, x2),
and(x2, nand(or(x3, x1), nand(x3, x1))))), x2))

```

A GBC generated program for 3-Multiplexer problem(absolute error=0)

```

and(nand(x1, x2), or( and(x3, nand(or( and(x3, x1), x2), and(x2, x3))),
or(nor(nor(x2, x2), and(x3, and(nand(x1, x3), x3))), x1)))

```

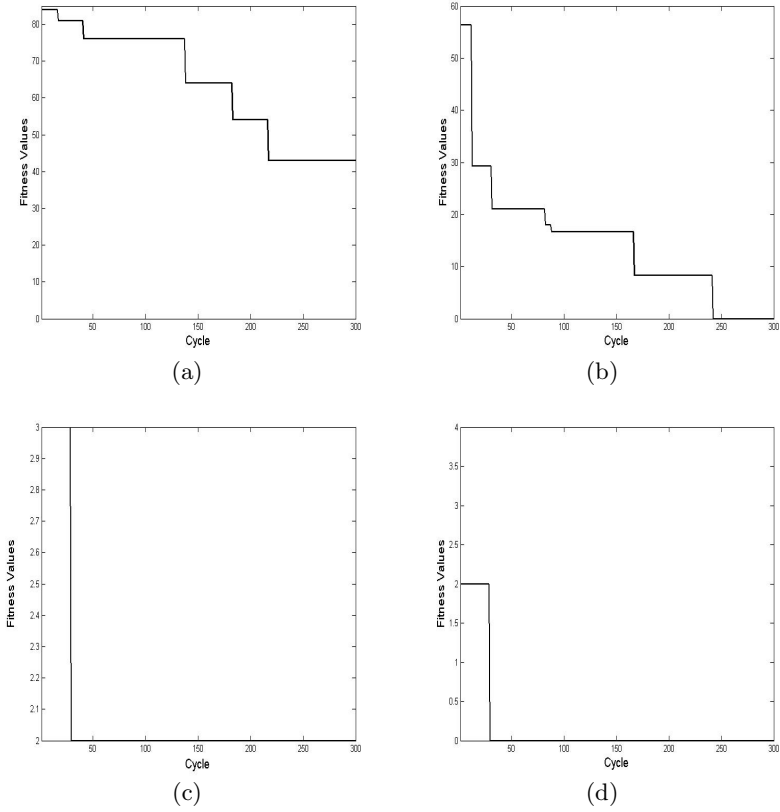


Fig. 2. Convergence graphs of GBC for Santa Fe Ant Trail(a), Symbolic Regression(b), Even-3 Parity(c), Multiplexer(d)

5 Conclusions

Grammatical Bee Colony algorithm is devised in this paper. Grammatical Bee Colony is a variant of Grammatical Evolution in which Artificial Bee Colony algorithm is used as search engine to generate computer programs by mapping from genotype to phenotype. The Grammatical Bee Colony algorithm is applied to the benchmark problems and compared with Grammatical Differential Evolution, Grammatical Swarm algorithm. Experiment results shows that, like Grammatical Differential Evolution and Grammatical Swarm, Grammatical Bee Colony is also able to generate computer programs. Future work will lead to performance improvement of Grammatical Bee Colony algorithm as well as its applications. A comparative study with Artificial Bee Colony Programming can be an another interesting future work of this paper.

References

1. Ryan, C., Collins, J.J., O'Neill, M.: Grammatical Evolution: Evolving Programs for an Arbitrary Language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds.) EuroGP 1998. LNCS, vol. 1391, pp. 83–95. Springer, Heidelberg (1998)
2. O'Neill, M., Ryan, C.: Grammatical Evolution. *IEEE Trans. Evolutionary Computation* 5(4), 349–358 (2001)
3. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press (1992)
4. Nohejl, A.: Grammar-based genetic programming. Master thesis, Charles University, Prague (2011)
5. O'Neill, M., Brabazon, A.: Grammatical Swarm: The Generation of Programs by Social Programming. *Natural Computing* 5(4), 443–462 (2006)
6. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, Perth, Australia (1995)
7. O'Neill, M., Brabazon, A.: Grammatical Differential Evolution. In: International Conference on Artificial Intelligence (ICAI 2006), pp. 231–236. CSEA Press, Las Vegas (2006)
8. Karaboga, D.: An Idea Based On Honey Bee Swarm For Numerical Optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
9. Karaboga, D., Ozturk, C., Karaboga, N., Gorkemli, B.: Artificial Bee Colony Programming for Symbolic Regression. *Information Sciences* 209, 1–15 (2012)
10. Poli, R., Langdon, W.B., McPhee, N.F.: A Field Guide to Genetic Programming (2008), Published via <http://lulu.com>, and freely available at <http://lulu.com>, (With contributions by J.R Koza). *GP_{BIB}*
11. Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: IEEE World Congr. Comput. Intell., pp. 69–73 (1998)

Artificial Bee Colony Algorithm for Probabilistic Target Q-coverage in Wireless Sensor Networks

S. Mini¹, Siba K. Udgata², and Samrat L. Sabat³

¹ Department of Computer Science and Engineering
School of Engineering and Technology
Central University of Rajasthan, Rajasthan, India
mini2min2002@yahoo.co.in

² School of Computer and Information Sciences
University of Hyderabad, Hyderabad-500046, India
udgatacs@uohyd.ernet.in

³ School of Physics
University of Hyderabad, Hyderabad-500046, India
slssp@uohyd.ernet.in

Abstract. The lifetime of a wireless sensor network is dependent on the type of sensor deployment. If the application permits deterministic deployment of nodes and if the sensor nodes are limited, quality of sensing and energy conservation can be enhanced by restricting the sensing range requirement. This paper addresses deterministic deployment of nodes for probabilistic target Q-coverage. A probabilistic coverage model considers the effect of distance and medium on the sensing ability of a node. We use Artificial Bee Colony (ABC) algorithm to compute the optimal deployment of sensor nodes such that the required sensing range is minimum for probabilistic target Q-coverage.

Keywords: Sensor Deployment, Target Coverage, Q-Coverage, ABC Algorithm.

1 Introduction

Maximization of network lifetime is one of the main challenges in wireless sensor networks. Energy can be efficiently used through proper energy utilization schemes, depending on the application. If the application permits deterministic deployment of nodes, energy wastage can be controlled by restricting the sensing range required and the quality of coverage can be improved in case of probabilistic coverage.

Sensor nodes are deployed to achieve either area coverage or target coverage. To achieve area coverage, if the sensors have fixed sensing range, optimal deployment patterns are preferred to minimize the number of sensor nodes required. The coverage requirement of the application also has an impact on the network lifetime. Some applications require that each target has to be monitored by at least one sensor node (simple coverage), whereas some might require a high number of sensor nodes to monitor the targets (k -coverage). In some applications, each target may

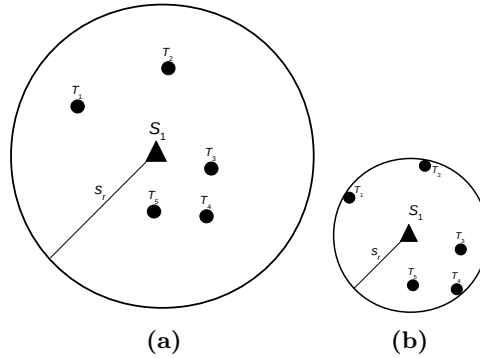


Fig. 1. Comparison (a) Random deployment with fixed sensing range and (b) Deterministic deployment with optimal sensing range

have a different coverage requirement. This is known as **Q**-coverage problem. In this paper, we focus on probabilistic target **Q**-coverage problem. The number of sensor nodes to be deployed are limited and the sensing range has to be minimized to control energy usage and improve quality of coverage.

Figure 1a shows a random deployment of sensor node S_1 . Here S_1 has a pre-defined sensing range s_r . Figure 1b shows a case of deterministic deployment of S_1 where the sensing range is restricted/reduced so that it is optimal for all the targets to be monitored.

1.1 Types of Sensor Deployments

The deployment of sensor networks varies with the application considered. In some environments, it can be predetermined and be placed in the exact locations. For some environments, the nodes can be air-dropped or deployed by other means [1].

Depending on the density of nodes in a network, a sensor network deployment can be categorized as dense deployment or sparse deployment.

– Dense Deployment

A dense deployment involves relatively large number of sensor nodes. It is used when higher level of coverage has to be satisfied.

– Sparse Deployment

A sparse deployment includes only a few number of nodes. It is used when dense deployment is not feasible because of cost of deployment or other factors.

Based on the type of deployment, a sensor network deployment can be categorized as random deployment or deterministic deployment.

- Random Deployment

Random deployment is suitable if no prior knowledge of the region is available. It is mainly used for military applications, inaccessible area, hostile region etc. However, random deployment does not always lead to effective coverage, especially if the sensors are clustered at some parts of the region [2].

- Deterministic Deployment

Deterministic deployment is suitable for accessible regions. It is also preferred if powerful, sophisticated and expensive nodes are used which require careful planning and placement [3]. Non-sensitive applications usually use deterministic deployment.

1.2 Types of Sensing Models

Most research works assume that the sensing region of a sensor node is a sensing disc, that is, a sensor node has the uniform contribution in all directions of its sensing region. In the basic (binary) model, if a target lies within the sensing region of a sensor node, it is always assumed to be detected with probability 1 otherwise with probability 0. This idealized binary model has been extensively used to analyze the coverage problems of sensor networks. But in real deployment of sensor nodes, the sensing capabilities of sensor nodes have relations with the environment and then it is imperative to have practical considerations at the design stage. Such a sensing model is known as probabilistic sensing model. Thus, in general there are two sensing models: binary sensing model and probabilistic sensing model.

Binary Sensing Model In a binary sensing model, the target is either monitored with full confidence or not monitored. Let $S = \{S_1, S_2, \dots, S_m\}$ be the set of sensor nodes and $T = \{T_1, T_2, \dots, T_n\}$ 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 s_r .

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \leq s_r \quad (1)$$

A binary sensing model is given by,

$$ST_{ij} = \begin{cases} 1 & \text{if } d_{ij} \leq s_r, \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$. d_{ij} corresponds to the Euclidean distance between S_i and T_j

Probabilistic Sensing Model. The sensing range is not a disk in probabilistic sensing models [4]. With probabilistic model, the probability that the sensor

detects a target depends on the relative position of the target within the sensors' sensing range. Basically the probability of detecting a target is assumed to diminish at an exponential rate with the increase in distance between a sensor and that target. 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. [5].

As in [6], we use the following exponential function to represent the confidence level in the received sensing signal:

$$ST_{ij} = \begin{cases} e^{-\alpha d_{ij}} & \text{if } d_{ij} \leq s_r, \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

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_j which is monitored by multiple sensor nodes S_j is given by,

$$ST_j(S_j) = 1 - \prod_{S_i \in S_j} (1 - ST_{ij}) \quad (4)$$

2 Related Work

Network performance metrics such as energy consumption, coverage, delay and throughput etc. are affected by the position of sensor nodes. For example, large distances between nodes weaken the communication links, lower the throughput and increase the energy consumption [6]. It is always a challenging task to make use of the available energy in a fair manner. The coverage requirement varies from application to application. Q-coverage is one where each target may have a different coverage requirement. Earlier work on Q-coverage problem [7][8][9] is based on random deployment of nodes. The model under consideration is binary. In such cases, sensor scheduling is a solution to maximize the network lifetime. The method proposed by Gu et al. [7] is based on column generation, where each column corresponds to a feasible solution. Chaudhary et al. [8] present a greedy heuristic, High Energy and Small Lifetime (HESL), to generate Q-covers by prioritizing sensors in terms of the residual battery life. Liu et al. [9] also propose a heuristic for Q-coverage problem with scheduling as a solution.

Du et al. [10] propose to improve sensor network performance by deploying some mobile sensors in addition to a large number of static sensors. Mobile sensors are used to increase sensing coverage, provide better routing and connectivity for sensor networks. The areas that require better coverage are identified and mobile sensors are moved towards that area. Shen et al. [11] propose Grid Scan which is applied to calculate the basic coverage rate with arbitrary sensing radius of each node. This approach is used to ensure k -coverage of the area and it is used to provide better coverage with less nodes. A re-deployment approach is used to obtain better coverage rate.

Initial work on probabilistic coverage address area coverage problem [5][12]. In [12], the problem is formulated as an optimization problem with the objective to minimize cost while all the points are covered with some required probability. The minimum number of sensors that are to be activated to maintain the required level of coverage is identified in [4].

Artificial Bee Colony (ABC) Algorithm [13] is an optimization algorithm based on the intelligent foraging behavior of honey bee swarm. ABC algorithm has been found to solve optimization problems efficiently. There are other algorithms also based on the behavior of natural bees [14]. Compared to other swarm intelligence based algorithms like GA, Particle Swarm Optimization (PSO), ABC is observed to perform better [13]. ABC algorithm was applied to sensor deployment problem in irregular terrain by Udgata et al. [15]. Mini et al. used ABC algorithm initially to solve simple coverage problem [16] and later to solve k -coverage and \mathbf{Q} -coverage problems [17]. The sensing model used in these problems is a binary sensing model, where a target is fully monitored or not at all monitored.

The dynamic deployment problem in WSNs with mobile sensors on a binary sensing model was solved using ABC algorithm [18]. Ozturk et al. [19] use ABC algorithm to solve dynamic deployment problem in WSNs within the scenario of mobile and stationary sensors on a probabilistic detection model. Andersen et al. [20] consider sensor deployment in a three dimensional space to achieve a desired degree of coverage and also to minimize the number of sensor placed. Most of the existing works on sensor deployment address area coverage problem in wireless sensor networks.

To the best of our knowledge, this is the first work to address sensor deployment for probabilistic target \mathbf{Q} -coverage problem. This paper is an extension to our earlier work [21] where we use artificial bee colony algorithm to solve probabilistic target k -coverage problem.

3 Problem Definition

Given a set of n targets $T = \{T_1, T_2, \dots, T_n\}$ located in $U \times V \times W$ region and m sensor nodes $S = \{S_1, S_2, \dots, S_m\}$, place the nodes such that $T = \{T_1, T_2, \dots, T_n\}$ is monitored by $\mathbf{Q} = \{q_1, q_2, \dots, q_n\}$ number of sensor nodes such that target T_j needs to be monitored by at least q_j number of sensor nodes, where $1 \leq i \leq n$ and with a total probability p such that the required sensing range is minimum. The objective is to cover T_j with at least q_j sensor nodes, probability p and to minimize

$$F = \forall_i((\max(\text{distance}(S_i, H_g)))) \quad (5)$$

where H is the set of all targets monitored by S_i , $i = 1, 2, \dots, m$, $g = 1, 2, \dots, h$, where h is the total number of targets S_i monitors.

Algorithm 1. Proposed Method of sensor deployment for probabilistic sensing model

```

1: Initialize the solution population  $B$ 
2: Evaluate fitness ((Equation(5)))
3: Produce new solutions based on probabilistic Q-coverage
4: Choose the fittest bee
5:  $cycle = 1$ 
6: repeat
7:   Search for new solutions in the neighborhood
8:   if new solution better than old solution then
9:     Memorize new solution and discard old solution
10:  end if
11:  Replace the discarded solution with a newly randomly generated solution
12:  Memorize the best solution
13:   $cycle = cycle + 1$ 
14: until  $cycle = maximumcycles$ 

```

4 Proposed Method

Let B be the solution population in a region with stationary targets. Each solution $B_a = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_m, y_m, z_m)\}$ where $a = 1, 2, \dots, nb$, nb the total number of bees and m 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 Q-coverage. Let R_j be the subset of sensor nodes which can make each target T_j meet the required probability. If R_j satisfies Q-coverage requirement of T_j , T_j is assigned to each sensor node in R_j . If it does not satisfy Q-coverage, then identify the nearest nodes which do not belong to R_j that can make T_j Q-covered, along with R_j . T_j is assigned to these new sensor nodes in addition to R_j . Each sensor node is then placed at the center of all the targets which are assigned to it. If some target will not be probabilistically covered due to this shift of location, this move should not be allowed.

The fitness function used to evaluate the solutions is the euclidean distance between each target and the sensor location to which it is associated. Each sensor node is associated to a cluster, where a cluster corresponds to the set of targets monitored by the sensor node. Let $D_i = (D_{i1}, D_{i2}, D_{i3})$ be the initial position of i^{th} cluster. $F(D_i)$ refers to the nectar amount at food source located at D_i . After watching the waggle dance of employed bees, an onlooker goes to the region of D_i with probability G_i defined as,

$$G_i = \frac{F(D_i)}{\sum_{l=1}^m F(D_l)} \quad (6)$$

where m is the total number of food sources. The onlooker finds a neighborhood food source in the vicinity of D_i by using,

$$D_i(t+1) = D_i(t) + \delta_{ij} \times v \quad (7)$$

Table 1. Sensing Range for Probabilistic Target **Q**-Coverage

| α | Probability | Q | Sensing Range | | |
|----------|-------------|--------|---------------|--------|--------------------|
| | | | Best | Mean | Standard Deviation |
| 0.05 | 0.6 | 1-2 | 2.0616 | 2.0616 | 0 |
| | | 1-3 | 3.9751 | 4.0235 | 0.0439 |
| | | 1-4 | 3.9427 | 3.9547 | 0.0208 |
| | 0.7 | 1-2 | 2.0616 | 2.0616 | 0 |
| | | 1-3 | 3.7723 | 3.8672 | 0.0836 |
| | | 1-4 | 3.9147 | 3.9761 | 0.0719 |
| | 0.8 | 1-2 | 2.0616 | 2.0616 | 0 |
| | | 1-3 | 3.8876 | 3.9565 | 0.0815 |
| | | 1-4 | 3.8286 | 3.9514 | 0.1233 |
| 0.9 | 1-2 | 2.0616 | 2.0616 | 0 | |
| | 1-3 | 3.943 | 3.9721 | 0.0297 | |
| | 1-4 | 3.9254 | 3.9858 | 0.055 | |
| 0.1 | 0.6 | 1-2 | 2.0616 | 2.0616 | 0 |
| | | 1-3 | 3.8579 | 3.9443 | 0.1061 |
| | | 1-4 | 3.9178 | 3.9591 | 0.0611 |
| | 0.7 | 1-2 | 2.0616 | 2.0616 | 0 |
| | | 1-3 | 3.8544 | 3.9946 | 0.1216 |
| | | 1-4 | 3.8623 | 3.9178 | 0.1078 |
| | 0.8 | 1-2 | 2.0616 | 2.0616 | 0 |
| | | 1-3 | 3.9269 | 3.9494 | 0.0264 |
| | | 1-4 | 3.9652 | 4.0359 | 0.0627 |
| 0.9 | 1-2 | 3.6976 | 3.9003 | 0.1771 | |
| | 1-3 | 4.0306 | 4.1559 | 0.1095 | |
| | 1-4 | 4.1482 | 4.3142 | 0.1648 | |
| 0.15 | 0.6 | 1-2 | 2.0616 | 2.0616 | 0 |
| | | 1-3 | 4.0044 | 4.0378 | 0.0289 |
| | | 1-4 | 3.9331 | 3.9378 | 0.004 |
| | 0.7 | 1-2 | 2.0616 | 2.0616 | 0 |
| | | 1-3 | 3.9766 | 3.9827 | 0.0061 |
| | | 1-4 | 3.9273 | 3.9803 | 0.0912 |
| | 0.8 | 1-2 | 3.4953 | 3.6227 | 0.1207 |
| | | 1-3 | 3.8904 | 4.0533 | 0.168 |
| | | 1-4 | 3.9161 | 4.0423 | 0.1204 |
| 0.2 | 0.6 | 1-2 | 2.0616 | 2.0616 | 0 |
| | | 1-3 | 3.9354 | 3.9735 | 0.0449 |
| | | 1-4 | 3.8028 | 3.9391 | 0.1228 |
| | 0.7 | 1-2 | 3.4768 | 3.5369 | 0.0941 |
| | | 1-3 | 3.9114 | 4.0044 | 0.1294 |
| | | 1-4 | 3.8978 | 3.9552 | 0.0516 |

The onlooker finds a neighborhood food source in the vicinity of D_i by using Equation 7. The solutions are never 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, that solution is chosen and the old one

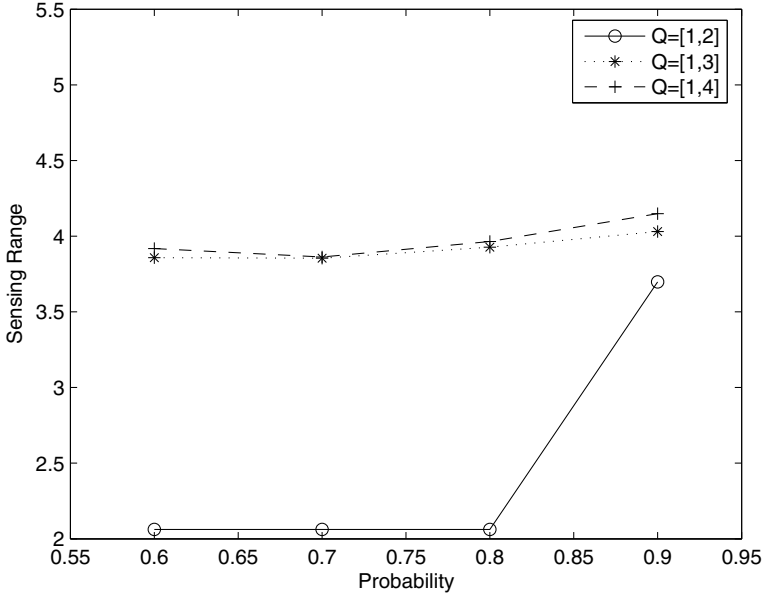


Fig. 2. Sensing range requirement for 10x10x10 grid with $\alpha=0.1$

is discarded. Scout bees search for a random feasible solution. The solution with the least sensing range and that satisfies probabilistic \mathbf{Q} coverage is finally chosen as the best solution. The proposed scheme is shown in Algorithm 1.

5 Results and Discussion

The initial experiments are carried out on a 10x10x10 grid. 5 sensor nodes are to be deployed to monitor 10 targets. The total number of bees (colony size) is 10, with half of them being employed bees and half onlookers. The number of cycles is 500, limit for neighborhood search is 20 and the number of runs is 3. MATLAB 2007a is used for implementation.

Since a probabilistic coverage model considers the effect of medium on the sensing ability of a node, we vary the value of α . α is assumed to take values 0.05, 0.1, 0.15 and 0.2. The value of \mathbf{Q} also affects the sensing range required. We vary \mathbf{Q} -values 1 to 2, 1 to 3 and 1 to 4. The required probability is set to 0.6, 0.7, 0.8 and 0.9. The sensing range required depends highly on \mathbf{Q} . With α at 0.15, the targets cannot be covered with a probability 0.9 and with $\alpha = 0.2$, the targets cannot be covered with a probability 0.8 or higher (Table 1). Figure 2 shows the sensing range requirement for this set-up with $\alpha = 0.1$. It can clearly be seen that for the targets to be monitored with $\mathbf{Q}=[1,2]$, the sensing range required for coverage with probability 0.6, 0.7 and 0.8 are the same. But when a higher probability of 0.9 is required, a noticeable increase in the sensing range is observed. When the

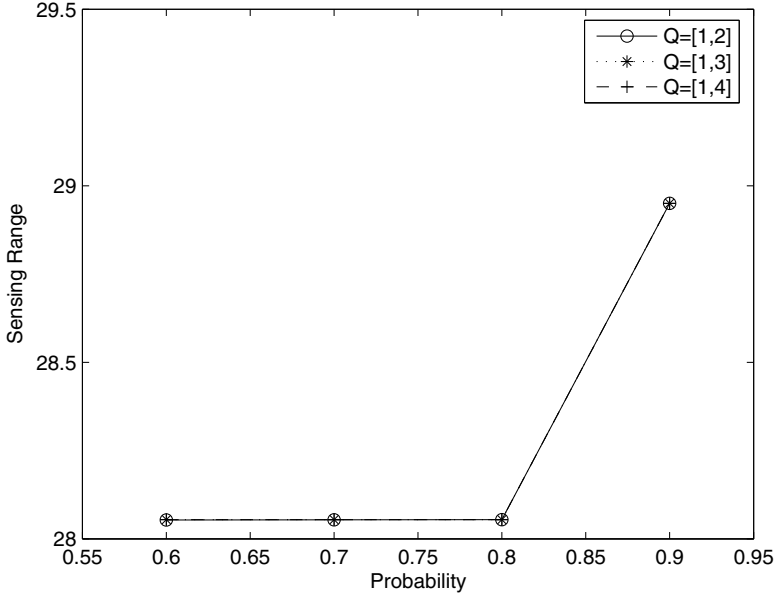


Fig. 3. Sensing range requirement for 100x100x20 grid

Table 2. Sensing Range for Probabilistic Target **Q**-coverage (100x100x20 grid)

| Probability | Q | Sensing Range | | |
|-------------|-----|---------------|---------|--------------------|
| | | Best | Mean | Standard Deviation |
| 0.6 | 1-2 | 28.0531 | 28.0764 | 0.0378 |
| | 1-3 | 28.0543 | 28.0998 | 0.0517 |
| | 1-4 | 28.0545 | 28.0560 | 0.0027 |
| 0.7 | 1-2 | 28.0538 | 28.0955 | 0.0422 |
| | 1-3 | 28.0545 | 28.1372 | 0.0737 |
| | 1-4 | 28.0545 | 28.1345 | 0.1386 |
| 0.8 | 1-2 | 28.0545 | 28.4431 | 0.3564 |
| | 1-3 | 28.0545 | 28.4431 | 0.3564 |
| | 1-4 | 28.0545 | 28.4431 | 0.3564 |
| 0.9 | 1-2 | 28.9500 | 29.2468 | 0.2888 |
| | 1-3 | 28.9500 | 29.2468 | 0.2888 |
| | 1-4 | 28.9500 | 29.2468 | 0.2888 |

value of **Q** is increased, the sensing range required for coverage with probability 0.6, 0.7 and 0.8 is much higher compared to that for **Q**=[1,2]. But there is no major change in the sensing range requirement even for a higher probability requirement of 0.9. The measured standard deviation shows that the method is a robust one.

We also consider a $100 \times 100 \times 20$ grid for experimentation to see how the algorithm performs on a larger grid. Three instances of 100 targets being monitored by 10 sensor nodes are considered. The results are reported as an average of the sensing range required for these three instances. α is assumed to be 0.01. Figure 3 clearly shows that there is no significant change in the sensing range requirement for different values of Q . The slight variation in the sensing range required can be seen in Table 2. With Q -coverage requirement, for probability 0.8 and 0.9, the sensing range required for Q values 1 to 2, 1 to 3 and 1 to 4 are the same. These results show that the algorithm is a reliable one even for a larger grid.

6 Conclusion

In this paper, we consider deterministic deployment of sensor nodes for probabilistic sensing model where the deployment locations are computed such that the required sensing range is minimum and probabilistic Q -coverage requirement is satisfied. Here, the number of sensor nodes is assumed to be limited. ABC algorithm is used to compute the optimal deployment locations. We studied the variation in sensing range for a range of detection probabilities (p), coverage requirement (Q) and physical medium characteristics (α). The use of ABC algorithm proves to be a reliable one since no significant change is observed in the standard deviation of obtained sensing range among various runs for a larger region or for higher values of Q . In future, we plan to study connected coverage with probabilistic detection model.

References

1. Poe, W.Y., Schmitt, J.B.: Node Deployment in Large Wireless Sensor Networks: Coverage, Energy Consumption, and Worst-Case Delay. In: Proc. of Asian Internet Engineering Conference, pp. 77–84 (2009)
2. Zou, Y., Chakrabarty, K.: Sensor Deployment and Target Localization Based on Virtual Forces. In: Proc. of INFOCOM, pp. 1293–1303 (2003)
3. Liu, X., Mohapatra, P.: On the Deployment of Wireless Sensor Nodes. In: Proc. of Third International Workshop on Measurement, Modelling, and Performance Analysis of Wireless Sensor Networks, in Conjunction with MobiQuitous (2005)
4. Hefeeda, M., Ahmadi, H.: Energy-Efficient Protocol for Deterministic and Probabilistic Coverage in Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems* 21, 579–593 (2010)
5. Ahmed, N., Kanhere, S., Jha, S.: Probabilistic Coverage in Wireless Sensor Networks. In: Proc. of IEEE Conference on Local Computer Networks, pp. 672–681 (2005)
6. Zou, Y., Chakrabarty, K.: A Distributed Coverage- and Connectivity-Centric Technique for Selecting Active Nodes in Wireless Sensor Networks. *IEEE Transactions on Computers* 54, 978–991 (2005)
7. Gu, Y., Liu, H., Zhao, B.: Target Coverage With QoS Requirements in Wireless Sensor Networks. In: Proc. of Intelligent Pervasive Computing, pp. 35–38 (2007)

8. Chaudhary, M., Pujari, A.K.: Q-Coverage Problem in Wireless Sensor Networks. In: Garg, V., Wattenhofer, R., Kothapalli, K. (eds.) ICDCN 2009. LNCS, vol. 5408, pp. 325–330. Springer, Heidelberg (2008)
9. Liu, H., Chen, W., Ma, H., Li, D.: Energy-Efficient Algorithm for the Target Q-coverage Problem in Wireless Sensor Networks. In: Pandurangan, G., Anil Kumar, V.S., Ming, G., Liu, Y., Li, Y. (eds.) WASA 2010. LNCS, vol. 6221, pp. 21–25. Springer, Heidelberg (2010)
10. Du, X., Lin, F.: Improving Sensor Network Performance by Deploying Mobile Sensors. In: Proc. of 24th IEEE International Performance, Computing, and Communications Conference (2005)
11. Shen, X., Chen, J., Sun, Y.: Grid Scan: A Simple and Effective Approach for Coverage Issue in Wireless Sensor Networks. In: Proc. of IEEE International Conference on Communications (2006)
12. Carter, B., Ragade, R.: A Probabilistic Model for the Deployment of Sensors. In: Proc. of IEEE Sensors Applications Symposium, pp. 7–12 (2009)
13. 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)
14. Mozaffari, A., Gorji-Bandpy, M., Gorji, T.: Optimal design of constraint engineering systems: application of mutable smart bee algorithm. *Int. J. of Bio-Inspired Computation* 4(3), 167–180 (2012)
15. 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)
16. 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)
17. 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)
18. Ozturk, C., Karaboga, D., Gorkemli, B.: Artificial Bee Colony Algorithm for Dynamic Deployment of Wireless Sensor Networks. *Turkish Journal of Electrical Engineering & Computer Sciences* 20, 255–262 (2012)
19. Ozturk, C., Karaboga, D., Gorkemli, B.: Probabilistic Dynamic Deployment of Wireless Sensor Networks by Artificial Bee Colony Algorithm. *Sensors* 11, 6056–6065 (2011)
20. Andersen, T., Tirthapura, S.: Wireless Sensor Deployment for 3D Coverage with Constraints. In: Proc. of Sixth International Conference on Networked Sensing Systems, INSS (2009)
21. Mini, S., Udgata, S.K., Sabat, S.L.: Sensor Deployment for Probabilistic Target k-Coverage using Artificial Bee Colony Algorithm. In: Panigrahi, B.K., Suganthan, P.N., Das, S., Satapathy, S.C. (eds.) SEMCCO 2011, Part I. LNCS, vol. 7076, pp. 654–661. Springer, Heidelberg (2011)

Chaos Synchronization in Commensurate Fractional Order Lü System via Optimal $PI^\lambda D^\mu$ Controller with Artificial Bee Colony Algorithm

Anguluri Rajasekhar¹, Shantanu Das², and Swagatam Das³

¹ Dept. of Electrical Engineering, National Institute of Technology-Warangal, 506004, India

² Reactor Control Division, Bhabha Atomic Research Center, Mumbai-400085, India

³ Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, India

{rajasekhar.anguluri, swagatam.das}@ieee.org,
shantanu@barc.gov.in

Abstract. In this paper a master-slave approach to chaos synchronization in commensurate fractional order (FO) Lü system has been studied via optimally designed fractional order Proportional Integral Derivative (PID) Controller. An optimization frame work based on integral time indices has been considered for tuning the unknown parametric constants of FOPID controller. The optimization task was carried out using bees swarm intelligent based Artificial Bee Colony algorithm. A comparative study based on conventional PID control scheme has also been performed to highlight the advantage of using fractional order controller for fractional order systems. Simulation results presented for synchronization of chaos in two Lü systems supports the claim for proposed approach.

Keywords: Attractor, Chaos, Lyapunov exponent, artificial bee colony, swarm intelligence.

1 Introduction

In a class of nonlinear dynamical systems, chaotic behavior can be found in systems that show different trajectories in the phase portrait for different initial conditions. Among many definitions for a system to exhibit chaos, the most widely accepted is that of Lyapunov exponents (corresponding to the states of the governing differential equation should be positive) [1]. In real time applications like secure communications, biological systems, information processing [2], synchronization of such chaotic systems with unequal initial conditions is necessary. Since last decade the concept of fractional calculus attracted researchers working in the area of physics and engineering applications. Though it is a 300 year old topic, it found its application in late 1900. Many systems such as visco-elastic systems, electromagnetic waves etc., are known to exhibit fractional dynamics [3]. Recent studies also revealed that chaos exist in nonlinear systems those exhibit fractional order dynamics [4].

Chaotic behaviors have been found in many systems of them which include Lorenz system, Chen system, Lü system, Rossler system etc [4]. Lü system acts as a bridge between Lorenz and Chen system and it represents transition from one to other. In this paper we considered chaos synchronization in two identical fractional order Lü systems governed by two different initial conditions. Synchronization has been done with the aid of external control action. Since the chaotic systems' trajectories will deviate widely even for minute changes in initial conditions, external control action should be employed for faster synchronization. As the system considered has fractional characteristics it is wise to use a controller having the similar characteristics. Hence we propose fractional order PID or $PI^\lambda D^\mu$ controller for synchronization purpose. The concept of $PI^\lambda D^\mu$ controller has been invented by Podlubny [5], it has five degrees of freedom for tuning to meet additional design specifications for complicated systems.

A typical chaos synchronization problem can be considered as a master-slave or drive-response configuration. Traditionally integer order chaotic systems are implemented with aid of conventional PID controller, fuzzy controller etc [6, 7]. The same approach has been implemented with help of fractional components. As it has 5 unknown variables, designing an optimum FOPID controller requires fine tuning of parametric gains $\{K_p, K_i, K_d, \lambda, \mu\}$ which in return calls for real parameter optimization in five-dimensional hyperspace. Literature studies [17] and also our previous experiences [8] show the superior performance of recently evolved Artificial Bee Colony (ABC) algorithm for real world optimization problems. This superior performance can be ascribed to its well organized exploitation and exploration phases [8].

The rest of the paper is categorized as sections and is organized as follows. Section 2 gives a brief introduction to the integral (differential) operators used in fractional calculus and their approximations. Section 3 describes the master slave chaos synchronization framework with two FO Lü systems. In section 4 we discussed the designing principles of $PI^\lambda D^\mu$ and the objective function considered for tuning. Section 5 provides the essential rudiments of ABC algorithm. Section 6 presents the simulation results and comparisons with different controllers. Finally, this paper ends with conclusions and future scope in Section 7.

2 Fractional Derivatives and Integrals: A brief

A misnomer yet recently evolved as one of powerful tools in calculus i.e., Fractional Calculus is a generalization of differentiation and integration to a noninteger-order operator ${}_a D_t^\alpha$. This continuous integro-differential operator is defined as

$${}_a D_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha} & R(\alpha) > 0, \\ 1 & R(\alpha) = 0, \\ \int_a^t (d\tau)^{-\alpha} & R(\alpha) < 0. \end{cases} \tag{1}$$

Where $R(\alpha)$ represents the real part of fractional order. There are many definitions for general fractional differentiation and can be found in literature [4, 9]. The three most popular and frequently used definitions are the Grünwald-Letnikov (GL) definition, the Riemann-Liouville (RL) and Caputo definitions. It was also shown that for wide range of functions, these GL, RL and Caputo are equivalent under some conditions [10]. In this article we considered GL based definition for time based simulation of FO chaotic Lü system. Further we also used Caputo definition for developing the controller for synchronization of chaotic systems (which will be explained in section 3). According to GL based definition for α^h order integro-differential of a function can be written as

$${}_0D_t^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{\lfloor \frac{t-a}{h} \rfloor} (-1)^j \binom{\alpha}{j} f(t - jh) \tag{2}$$

where $[x]$ represents the integer part of x , a and t are the bounds for ${}_aD_t^\alpha f(t)$. The binomial coefficients in Eqn (2) can be calculated using Euler’s *Gamma* function and factorial judiciously, defined as

$$\binom{\alpha}{j} = \frac{\alpha!}{j!(\alpha - j)!} = \frac{\Gamma(\alpha + 1)}{\Gamma(j + 1)\Gamma(\alpha - j + 1)} \tag{3}$$

An equivalent relation to the explicit numerical approximation of q -th order derivative at points kh ($k=1,2,\dots$) has the following form [4].

$$({}^{k-L_m}/h)D_k^q f(t) \approx \frac{1}{h^q} \sum_{j=0}^k c_j^{(q)} f(t_{k-j}) \tag{4}$$

where L_m is the “memory length”, $t_k = kh$, h is the time step and $c_j^{(q)}$ ($j=0,1,\dots$) are binomial coefficients. These can be recursively calculated with following formula [4]

$$c_0^{(q)} = 1, \quad c_j^{(q)} = \left(1 - \frac{1+q}{j}\right) c_{j-1}^{(q)} \tag{5}$$

The general numerical solution of the fractional differential equation is given by ${}_0D_t^q y(t) = f(y(t), t)$ and can be expressed as

$$y(t_k) = f(y(t_k), t_k)h^q - \sum_{j=1}^k c_j^{(q)} y(t_{k-j}) \tag{6}$$

Eqn (6) is used to evaluate the fractional derivatives involved in the Lü system.

3 Master-Slave Synchronization of Chaotic Fractional Order Lü Systems

According to [2], it is well known that a unified chaotic system may be represented by the following set of non-linear differential equations.

$$\left. \begin{aligned} \frac{dx}{dt} &= (25\alpha + 10)(y - x) \\ \frac{dy}{dt} &= (28 - 35\alpha)x - xz + (29\alpha - 1)y \\ \frac{dz}{dt} &= xy - \frac{\alpha + 8}{3}z \end{aligned} \right\} \tag{7}$$

where $0 \leq \alpha \leq 1$. If the extreme values of α are considered i.e., for $\alpha=0$ the system will be transformed to original Lorenz system and on the other hand for $\alpha=1$ the system represents the Chen system. Now for an intermediate value $\alpha=4/5$ the system is known as Lü system which connects the aforementioned Lorenz and Chen system and represents the transition from one to another. The following equations represent conventional Lü system

$$\frac{dx}{dt} = a(y - x); \quad \frac{dy}{dt} = (-xz + cy); \quad \frac{dz}{dt} = (xy - bz); \tag{8}$$

where $(a,b,c) \in R^3$ are constant parameters and when these set equals $(36, 3, 20)$ system (8) has a chaotic attractor [3]. Its generalization using fractional calculus with respect to time is given by Eqn (9)

$$\left. \begin{aligned} \frac{d^{q_1}x}{dt^{q_1}} &= {}_0D_t^{q_1}x(t) = a(y(t) - x(t)) \\ \frac{d^{q_2}y}{dt^{q_2}} &= {}_0D_t^{q_2}y(t) = -x(t)z(t) + cy(t) \\ \frac{d^{q_3}z}{dt^{q_3}} &= {}_0D_t^{q_3}z(t) = x(t)y(t) - bz(t) \end{aligned} \right\} \tag{9}$$

where $0 < q_1, q_2, q_3 \leq 1$ are the derivative orders for $x, y,$ and z respectively. The above states are numerically evaluated using Eqn (6) with help of Eqn (4) and (5) The system defined by Eqn (9) has three equilibrium points at

$$E_1 = (0;0;0); \quad E_2 = (\sqrt{bc} : \sqrt{bc}; c); \quad E_3 = (-\sqrt{bc} : -\sqrt{bc}; c)$$

The Eigen values for the Jacobian matrix (obtained via Eqn 9) evaluated at E_1, E_2, E_3 results in a saddle point and also two saddle-focus points [4]. Petras *et al.* suggested that, for the Eigen values obtained a minimal commensurate order to keep the system chaotic is $q > 0.915$. As this paper deals with commensurate order we considered $q_1=q_2=q_3=0.95$ and the obtained phase plots are depicted in Fig 1.

The master-slave configuration of fractional order Lü system is represented in (9) and (10). Assuming that the system defined in 9 represents master then corresponding slave system is given in Eqn (9).

$$\left. \begin{aligned} {}_0D_t^{q_1}\hat{x}(t) &= a(\hat{y}(t) - \hat{x}(t)) \\ {}_0D_t^{q_2}\hat{y}(t) &= -\hat{x}(t)\hat{z}(t) + c\hat{y}(t) + u(t) \\ {}_0D_t^{q_3}\hat{z}(t) &= \hat{x}(t)\hat{y}(t) - b\hat{z}(t) \end{aligned} \right\} \tag{10}$$

In Eqn (9), the control action $u(t)$ is the output of an optimally tuned fractional order PID controller and is given by

$$u(t) = K_p e(t) + K_I \frac{d^{-\lambda}}{dt^\lambda} e(t) + K_D \frac{d^{-\mu}}{dt^\mu} e(t)$$

Here $e(t)$ represents the error between state trajectory of any of the state of master and slave system for two different initial conditions. The following initial conditions have been considered for master and slave system respectively

$$\{x_0 = 0.2, y_0 = 0.5, z_0 = 0.3\} \rightarrow \text{Master system}$$

$$\{\hat{x}_0 = 20, \hat{y}_0 = 10, \hat{z}_0 = -5\} \rightarrow \text{Slave system}$$

4 Optimal Design of PID/FOPID Controller

4.1 Fractional Order Controller: A Brief

The idea of a fractional order PID or FOPID controller derives its origin from the concept of fractional order differentiation and integration [11, 12]. Though popular definitions of fractional derivative like Grunwald-Letnikov and Riemann Loville definitions are prevalent, in terms of fractional order systems Caputo definition is widely preferred [13]. This definition of fractional derivative is used to derive fractional order transfer function models from fractional order differential equations with zero initial conditions. According to Caputo’s definition the α^{th} order derivative of a function $f(t)$ with respect to time is given by following equation

$$D^\alpha f(t) = \frac{1}{\Gamma(m - \alpha)} \int_0^t \frac{D^m f(\tau)}{(t - \tau)^{\alpha+1-m}} d\tau \tag{11}$$

$$\alpha \in \mathbb{R}^+, m \in \mathbb{Z}^+, m - 1 \leq \alpha < m,$$

Laplace transform of Eqn (11) can be represented as

$$\int_0^\infty e^{-st} D^\alpha f(t) dt = s^\alpha F(s) - \sum_{k=0}^{m-1} s^{\alpha-k-1} D^k f(0) \tag{12}$$

where $\Gamma(\alpha) = \int_0^t e^{-t} t^{\alpha-1} dt$ is the Gamma function

$$F(s) = \int_0^\infty e^{-st} f(t) dt \text{ is Laplace transform of } f(t)$$

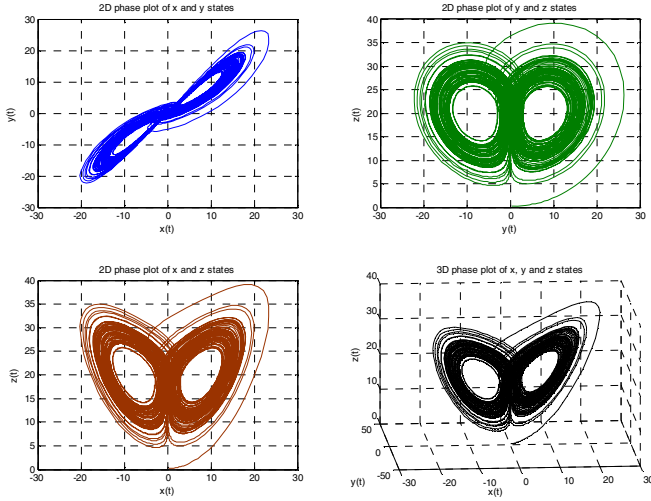


Fig. 1. Phase portraits among different state variables for the fractional order Chaotic Lü System

With an assumption of zero initial conditions the time domain operator D^α can be simply represented in frequency domain as s^α . A negative sign in the order of derivative ($-\alpha$) indicates a fractional integral operation. Hence the FOPID controller is a sum of fractional operators along with controller gains. The transfer function representation of a FOPID controller is given in Eqn (13)

$$C(s) = K_p + \frac{K_I}{s^\lambda} + K_D s^\mu \tag{13}$$

This typical controller consists of three controller gains $\{K_p, K_I, K_D\}$ and two more fractional order operators $\{\lambda, \mu\}$. For Instance, if $\lambda=1$ and $\mu=1$ Eqn (13) reduces to classical controller in parallel structure. In order to implement a controller of form Eqn (13), Oustaloup’s band limited frequency domain rational approximation technique has been used in this paper and also in most of FO control literatures [14].

4.2 Digital Realization of Fractional Orders

The rationale behind the choice of frequency domain rational approximation of FOPID controller is that it can be easily implemented in real hardware using higher order analog or digital filters, corresponding to each fractional order differentiation or integration in FOPID controller.

The infinite dimensional nature of fractional order differentiator and integrator in FOPID controller structure creates hardware implementation issues in industrial application of FOPID controllers. However, recent research results demonstrated that band-limited implementation of FOPID controllers using higher order rational transfer

function approximation of the integro-differential operators give satisfactory performance in industrial applications [15]. Oustaloup’s recursive approximation, which has been implemented to realize fractional integro-differential operators in frequency domain, is given by the following equation

$$s^\alpha = K \prod_{k=-N}^N \frac{s + \omega'_k}{s + \omega_k} \tag{14}$$

Here the poles, zeros and gain of the filter can be recursively evaluated as:

$$\omega_k = \omega_b \left(\frac{\omega_h}{\omega_b} \right)^{\frac{k+n+\frac{1}{2}(1+\alpha)}{2N+1}} ; \omega'_k = \omega_b \left(\frac{\omega_h}{\omega_b} \right)^{\frac{k+n+\frac{1}{2}(1-\alpha)}{2N+1}} \tag{15}$$

where $K = \omega_h^\alpha$. In above equation set α is the order of the differ-integration, $(2N+1)$ is the order of the filter and (ω_b, ω_h) is the expected fitting range. In the current study, 5th order Oustaloup’s recursive approximation is done for the integro-differential operators within a frequency band of the constant phase elements (CPEs) as $\omega \in \{10^{-2}, 10^2\}$ rad/sec.

4.3 Objective Function

PID/FOPID controller parameters are tuned in optimal fashion such that drive gives optimal performance. For tuning of controllers, we considered two objective functions i.e., Integral Absolute Error (IAE); Integral Time Squared Error (ITSE) criterion. The optimal parameters of *PID/FOPID* controller are obtained by minimizing these objective functions via an optimization algorithm. Eqn (16, 17) represents the mathematical formulation of these objective functions

$$J_1 = IAE = \int_0^\infty |e(t)| dt \tag{16}$$

$$J_2 = ITSE = \int_0^\infty t \cdot e^2(t) dt \tag{17}$$

Every integral performance index has certain advantages in control system design. The squared deviations have been minimized with help of an optimization algorithm. Also the time multiplication term in above Eqn (17) enforces the synchronization error to be small at later stages. The squared term in the error index puts large penalties for larger magnitudes of error than that with its absolute value resulting in high value of control signal.

5 Artificial Bee Colony Algorithm

Artificial Bee Colony (ABC) algorithm is a stochastic optimization algorithm, inspired by the foraging behavior of honey bees. ABC was first proposed by Karaboga [16, 17] for optimization of multivariable and multi-modal continuous functions. As of remaining evolutionary and swarm based algorithms ABC algorithm consists of two phases i.e., exploitation phase taken care by *employed & onlooker bee* and exploration phase taken care by *scout bee* [16]. In ABC algorithm, each solution corresponding to the problem is denoted as food source and is represented by a D -dimensional real-valued vector; on other hand fitness of solution corresponds to the nectar amount of associated food source. The algorithm begins by initializing all employed bees with randomly generated food sources (solutions). The position of i^{th} food source that corresponds to a solution in D -dimensional hyper space can be represented as $X_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$ and it can be generated by following equation.

$$x_{ij} = lb_j + rand(0,1) * (ub_j - lb_j) \quad (18)$$

Here, $i=1,2,3,\dots,FS$; $j=1,2,3,\dots,D$; FS is the number of food sources (equivalent to half to total number of bees) and D is the number of variables to be optimized; $rand$ is a random number in the range (0, 1); ub_j and lb_j corresponds to upper and lower bounds of the j^{th} dimension respectively.

Initially, an employed bee tries to exploit in the vicinity of random food source associated to it and updates its step based on Eqn (19)

$$x_{new} = x_{ij} + R \times (x_{ij} - x_{kj}) \quad (19)$$

where k is a randomly chosen index and $k \in (1,2,\dots,FS)$ such that $k \neq i$. R is a uniformly distributed random number in the range of $[-1, 1]$. As soon as x_{new} has been generated, a greedy mechanism is applied between x_{new} and its corresponding previous entity x_{ij} via fitness value. If the obtained new fitness value is better than the fitness value achieved so far, then the bee moves to this new food source discarding the old source. After all employed bees are done with local exploitation i.e, once employed bee phase is completed they share their information regarding food sources to *onlooker* bees. An onlooker bee selects a particular food source X_i based on the probability P_i defined as

$$P_i = \frac{fit_i}{\sum_{k=1}^{FS} fit_k} \quad (20)$$

fit_i corresponds to fitness value of i^{th} food source and as the chosen problem is a minimization problem fitness is calculated according to following equation.

$$fit_i = \frac{1}{(1 + J(X))} \quad (21)$$

$J(X)$ represents either J_1 or J_2 functions in this context. Based on above probability relation with respect to food source profitability onlooker tries to exploit a food source making use of Eqn (19) and similar greedy mechanism is performed. In the due course of iterative process, it may happen that a food source cannot be improved after N number of trials and this ultimately leads to delay in optimization process or leads to poor convergence. To eliminate this, an exploration scheme is been incorporated via scout bee. Each bee will search for a better food source for a certain number of cycles (*limit*), and if the fitness value does not improve then that particular bee becomes a Scout. Food source corresponding to that scout bee is abandoned and is initialized to random food source.

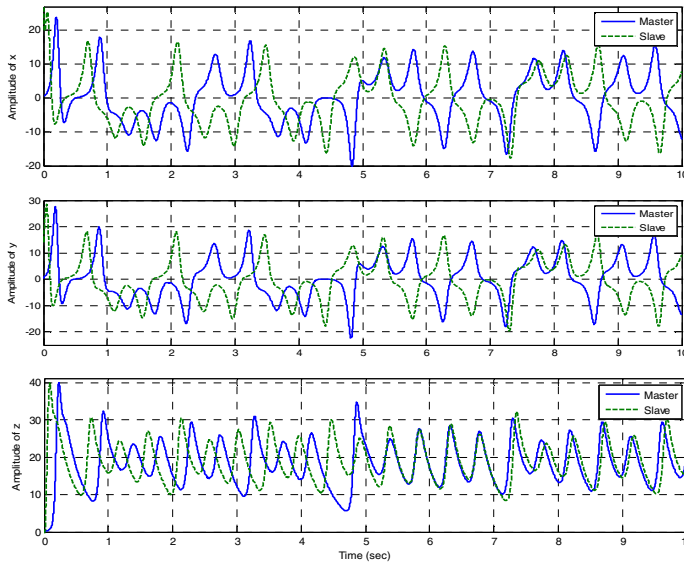


Fig. 2. Unsynchronized master and slave response of fractional order Lü systems

6 Simulations and Results

Fig. 2 depicts the responses of x, y, z of FO Lü system before synchronization. From the Fig 2 it is evident that for chosen initial conditions, master & slave response are showing wide deviations in their trajectories. This calls for design of good external controller for synchronization. To highlight the advantage of $PI^\lambda D^\mu$ controller over PID controller we also designed PID controller and compared it with the former.

6.1 J_1 Objective Function Minimization Based Design of PID/FOPID Controllers

J_1 represents Integral Absolute Error (IAE) based objective function. Table I provides the parametric gains obtained for PID/FOPID controller and also we recorded mean (standard deviation) for 25 independent runs. From the Table 1 and Fig 3 (a) it is clear

that FOPID provided less synchronization error when compared with PID based controller. Further from Fig 4 it is evident that with in less than 1 sec both the controllers are able to synchronize master response with slave response. On other hand a close look towards trajectory obtained in z state indicates that the synchronization was not good at earlier stages.

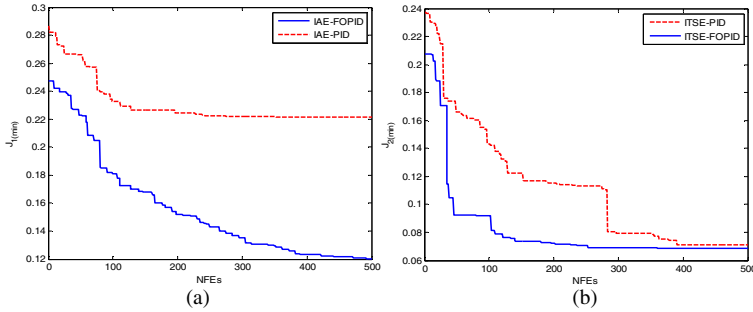


Fig. 3. Convergence of PID/FOPID for (a) IAE (b) ITSE based objective function towards optimum

6.2 J_2 Objective Function Minimization Based Design of PID/FOPID Controllers

J_2 denotes Integral time squared Error (ITSE) criterion. Table I provide the parametric gains obtained for PID and FOPID controller including mean (standard deviation) for 25 independent runs. In this approach also FOPID based design gave less objective function value than conventional PID controller. Controllers designed using this objective function criterion performed well to the previous one. Fig 5 shows the state trajectories for x, y, and z states of master and slave obtained using optimal PID and FOPID controllers.

From the figures it is evident that all the state trajectories are synchronized with in no time unless J_1 based design. Fig 3(b) shows the convergence of Artificial Bee colony algorithm towards minimum for ITSE based design. From Table 1, both J_1 and J_2 based design controllers indicate that there is less requirement of parameter gain K_D . This can be better visualized in the case of FOPID controller where K_D and associated derivative operator obtained to be 0. The main advantage of FOPID is that it gives more degree of freedom and hence tuning controller for a system considered may result in normal proportional control to complex FOPID controller.

Table 1. Comparison between IAE & ITSE based design

| | K_p | K_I | K_D | λ | μ | Mean | Std |
|------------|---------|--------|---------|-----------|-------|---------------|----------------|
| IAE-PID | 20.000 | 20.000 | 0.6902 | - | - | 0.2232 | 0.0118 |
| IAE-FOPID | 19.9341 | 20.000 | 19.2890 | 0.9293 | 0 | 0.1201 | 0.0016 |
| ITSE-PID | 9.0562 | 20.000 | 0 | - | - | 0.0731 | 0.0020 |
| ITSE-FOPID | 8.5789 | 20.000 | 0 | 0.9734 | 0 | 0.0684 | 1.55e-5 |

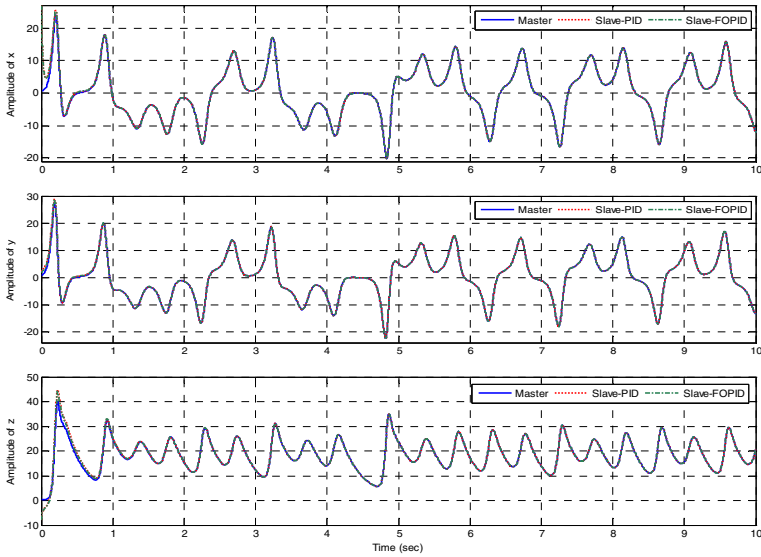


Fig. 4. Synchronization of the state trajectories for IAE based Controllers

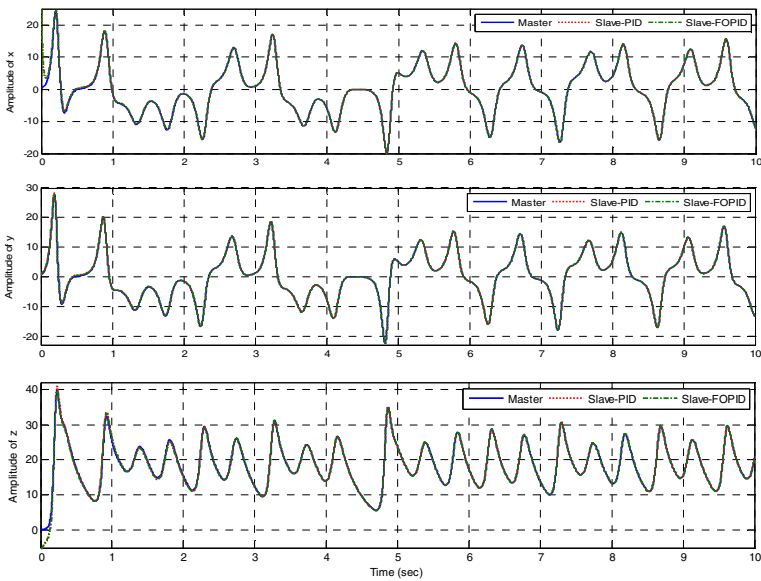


Fig. 5. Synchronization of the state trajectories for ITSE based Controllers

7 Conclusions

In this paper we propose an optimal fractional order controller design for chaos synchronization in two identical fractional order Lü systems. Swarm intelligence based

Artificial Bee Colony algorithm has been employed to obtain optimal parametric gains via minimizing an objective function. A conventional PID controller has also been designed and its performance is compared with proposed approach. From the simulations and results it was clear that FOPID controller outperformed PID controller, which suggest that a fractional system indeed require a fractional control signal. Our future research would focus on designing control schemes for chaos synchronization incommensurate systems and also for hyper chaotic systems.

References

1. Chen, A., Lu, J., Lu, J., Yu, S.: Generating “Generating hyperchaotic Lü attractor via state feedback control. *Physica A, Stat, Mech. Appl.* 364, 103–110 (2006)
2. Lu, J., Wu, X., Lu, J.: Synchronization of a unified chaotic system and the application in secure communications. *Phys. Lett. A* 305(6), 365–370 (2002)
3. Lu, J.G.: Chaotic dynamics of the fractional-order Lü system and its synchronization. *Phys. Lett. A* 354(6), 305–311
4. Petras, I.: *Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation, Non-linear Physical Science.* Springer (2011)
5. Podlubny, I.: Fractional-order systems and and $PI^{\lambda}D^{\mu}$ controllers. *IEEE Trans. Autom. Control.* 44(1), 208–214 (1999)
6. Wen, G., Wang, Q.-G., Li, C., Li, G., Han, X.: Chaos synchronization via multivariable PID control. *Int. J. Bifurc. Chaos* 17(5), 1753–1758
7. Li, S.-Y., Ge, Z.-M.: Generalized synchronization of chaotic systems with different orders by fuzzy logic constant controller. *Expert. Syst. Appl.* 38(3), 2302–2310 (2011)
8. Rajasekhar, A., Abraham, A., Pant, M.: Levy mutated Artificial Bee Colony algorithm for global optimization. In: 2011 International Conference on Systems, Man and Cybernetics (SMC 2011), pp. 655–662 (2011)
9. Podlubny, I.: *Fractional Differential Equations.* Academic Press, San Diego (1991)
10. Das, S.: *Functional Fractional Calculus, 2nd edn., Control Engineering.* Springer (2011)
11. Oldham, K.B., Spanier, J.: *The Fractional Calculus.* Academic Press, San Diego (1974)
12. Miller, K.S., Ross, B.: *An Introduction to the Fractional Calculus and Fractional Differential Equations.* Wiley, New York (1993)
13. Das, S., Pan, I., Das, S., Gupta, A.: Master-slave chaos synchronization via optimal fractional-order $PI^{\lambda}D^{\mu}$ controller with bacterial foraging algorithm. *Nonlinear Dynamics* 69(4), 2193–2206 (2012)
14. Valerio, D., da Costa, S.: Introduction to Single-input, Single-output fractional control. *IET Control Theory Appl.* 5(8), 1033–1057 (2011)
15. Effe, M.O.: Fractoinal order systems in industrial automation-a survey. *IEEE Tran. Ind. Inform.* 7(4), 582–591 (2011)
16. Karaboga, D., Basturk, B.: A Powerful and Efficient Algorithm for Numerical Optimization: Artificial Bee Colony (ABC) Algorithm. *Journal of Global Optim.* 3(39), 159–172 (2007)
17. Karaobga, D., Akay, B.: A Comprehensive survey: artificial bee colony algorithm and applications. *Artificial Intelligence Review* (2012)

Cooperative Micro Artificial Bee Colony Algorithm for Large Scale Global Optimization Problems

Anguluri Rajasekhar¹ and Swagatam Das²

¹Dept. of Electrical Engineering, National Institute of Technology-Warangal, India

²Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, India
{rajasekhar.anguluri, swagatam.das}@ieee.org

Abstract. Large scale optimization problems or optimization problems involving high-dimensions often appear in real world application scenario. The mathematical representation of these problems appears similar to that of traditional low dimensional problems but they exhibit high interdependencies among the variables to be optimized. Hence normal evolutionary algorithms or swarm intelligence based methods cannot be directly operated on these problems to find global optimum. In these situations, cooperating approaches are proved to be very valuable, since they are based on though simple yet power strategy “divide and conquer”. Though handy, computational burden of cooperative approach oriented methods will be high, as they involve optimization of various subcomponents for predefined number of steps. On other hand, recently evolved Micro Evolutionary Algorithms (micro-EAs) are shown to be very powerful strategies for solving optimization problems, as they involve very small population of just a few individuals. This advantage of micro-EA is accompanied by its tendency towards to get stuck in local optima. Hence this paper tries to combine the advantages of both cooperative strategies and also micro-EAs nature accompanied with a swarm intelligent Artificial Bee Colony (ABC) algorithm as main optimizer, to solve optimization problems of very high dimension. The proposed variant is termed as “Cooperative Micro-Artificial Bee Colony” (CMABC) algorithm. Computer simulations over benchmark suite considered and also extensive comparisons over cooperative variants of state-of-art Differential Evolution method show the superiority of proposed algorithm.

Keywords: cooperative approach, global optimization, large scale optimization.

1 Introduction

Curse of dimensionality is one of the serious problems in the field of optimization and often poses very difficult situation for normal evolutionary algorithm to tackle problems of high-dimension. Although many strategies for solving these kind of problems are initiated, propagated and developed, Cooperative Evolutionary Algorithms (CEAs) are proven to be one of best and powerful methods for solving intriguing large scale problems [1, 2]. CEAs follow a simple principle based on “divide and conquer” strategy, in which a problem of high-dimension is divided in to low-dimensional subcomponents. A huge number of cooperative populations attack these

low-dimensional components and evolve them concurrently [3]. Cooperation among subpopulations (populations that are used to evolve subcomponents) is responsible for exploring hidden landscapes and builds complete solution to the problem considered. CEA includes Genetic Algorithms (cooperative coevolutionary Genetic Algorithms) [2], Particle swarm optimization methods (CCPSO) [4, 5], Differential Evolution (COMDE) [3, 6], Artificial Bee Colony algorithms [7, 8]. Literature studies also reported various schemes that are to be considered while dividing dimensions, populations and they can be found in [5, 6].

On foil, another interesting class of algorithms frequently used for large-scale high dimension optimization problems consists of micro-EAs, which are instances of typical EAs characterized by small population size and often simple fitness. Initially they are applied to problems comprising dimensions in proportionate with their population size. Recently with help of some added heuristics they are also used to tackle very high dimension problems [9, 10, 11]. Search stagnation and attraction towards local optimum can be identified as main drawback of micro-EAs and it is primarily caused by their population size, which ultimately limits their exploration capability. Hence, efficiency is limited in high dimensions problem and becomes serious issue in complex and problems having strong interdependencies among variables. To overcome these kind of sprouting problems micro-EAs are usually combined with diversity-preserving schemes, so that the algorithm will not converge to same solution

To combine advantages of cooperative approaches and rapid convergence of micro-EAs we suggest a new algorithm termed as “Cooperative Micro Artificial Bee Colony” (CMABC) algorithm. Artificial Bee Colony algorithm is a newly evolved swarm intelligent (SI) algorithm extensively used for optimization problems ranging from numerical function optimization to real world problems [15]. This method has outperformed existing state of art evolutionary algorithms and remained to be one of the best contenders among EAs and SI algorithms. In this context, large scale problems involving very high dimensions are partitioned into low-dimensional subcomponents and each sub-component is handled by ABC algorithm employed with small population. Information sharing among subpopulations helps in constructing complete solutions for evaluation of each individual with the original objective function. A benchmark suite comprising of 5 test functions are considered to verify the performance of proposed approach and also CMABC is compared with conventional ABC method and also two cooperative variants of DE. Highly promising results over the compared methods followed by good convergence shows the superiority of proposed method.

The rest of paper is organized as follows: Section 2 deals with the rudiments of ABC algorithm. In Section 3 CMABC approach is unfold, followed by experimental results in Section 3. A brief discussions over the results obtained are provided in Section 5. The paper concludes while unfolding future research focus in Section 6.

2 Artificial Bee Colony Algorithm

Artificial Bee Colony (ABC) algorithm was first proposed and developed by Karaboga [12, 13] for solving numerical optimization problems [14]. Since its inception in 2005, ABC has been evolved in to a very powerful optimization tool in the field of swarm intelligence. As that of rest of swarm intelligent based methods like PSO,

ACO, BFO it is also a population-based stochastic optimization algorithm, which proceeds through iterations. Till now ABC remained one of the best competitors in optimizing various real world practical optimization problems ranging from 2-D PI controller problem to 150-D economic load dispatch problem. A comprehensive view of its developments and applications can be found in references cited [15].

This nature inspired method based on bees foraging strategy classifies bees into three groups namely *employed bees*, *onlooker bees* and *scout bees*. A colony of bees, say $2*n$ number of bees are equally distributed between employed and onlooker bees. A one-to-one mapping is associated between a food source and employed (onlooker) bee. An employed bee of abandoned food source becomes scout. ABC algorithm has a well constructed exploitation phase (carried out by employed and onlookers) and good exploration strategy (by scout). The underlying concept of ABC algorithm is summarized below.

1. Foragers equal to half of total number of bees are randomly employed to search the potential food sources.
2. A local exploitation will be initiated by a forager, if it visits a profitable food source and is termed as employed bee. It also retains position of food source.
3. After performing substantial amount of exploitation task, employed bees return to hive and then they share information about food sources with onlooker bees via waggle dances.
4. Based on dances (here equality profitability of food source) an onlooker chooses an employed bee food position. Now onlookers start to act as employed and employed bees (returned to hive) will be termed as onlookers.
5. Now the newly evolved employed bees start to exploit the potential areas. This process is repeated until all the onlookers are ready with new positions.
6. In due course of time it may happen that a source pertaining to an employed bee may be exhausted and eventually it is abandoned. The bee associated to abandoned source is termed as scout and it starts to explore new areas.

ABC algorithm begins by initializing all employed bees to randomly generated food sources (solutions) in a search space. In a n -dimensional hyper space, the position of a food source corresponding to i_{th} employed bee can be represented as $X_i = [x_{i1}, x_{i2}, \dots, x_{in}]$ and is generated by Eqn (1)

$$x_{ij} = lb_j + rand(0,1) \times (ub_j - lb_j) \tag{1}$$

Here, $i=1,2,3,\dots,FS$; $j=1,2,3,\dots,n$; FS is the number of food sources and n is the number of variables to be optimized; $rand$ is a random number in the range $(0, 1)$; ub_j and lb_j corresponds to upper and lower bounds of the j^{th} dimension respectively.

As discussed in Step 2, an employed bee tries to exploit the randomly visited food source and it updates its step using following Eqn.

$$x_{new} = x_{ij} + R \times (x_{ij} - x_{kj}) \tag{2}$$

where a randomly chosen index $k \in (1, 2, \dots, FS)$ such that $k \neq i$. R is a uniformly distributed random number in the range of $[-1, 1]$. Each employed bee performs greedy mechanism after updating its position. This mechanism is applied between x_{new} and its corresponding previous entity x_{ij} via fitness value. If the obtained new fitness value is better than the fitness value achieved so far, then the bee moves to this new food source discarding the old source. This will help the bees to retain best values every time and will drive the algorithm to converge much faster. After successful completion of exploitation task, employed bees share their information regarding food sources to *onlookers*. An onlooker bee selects a particular food source X_i based on the probability P_i defined as

$$P_i = \frac{fit_i}{\sum_{k=1}^{FS} fit_k} \quad (3)$$

fit_i corresponds to fitness value of i^{th} food source and as the chosen problem is a minimization problem fitness is calculated according to following equation.

$$fit_i = \frac{1}{1 + F(X)} \quad (4)$$

$F(X)$ corresponds to objective function to be minimized. Eqn (4) has been considered because the present objective of study is minimization and the algorithm has property of finding maximum profitable food source (solution). Hence Eqn (4) is used and for maximization directly fit_i represents objective function value. Based on above probability relation (3) with respect to food source profitability, onlooker tries to exploit a food source making use of Eqn (2) and similar greedy mechanism is performed. The above two phases i.e., employed bee and onlooker bee phases are performed in round robin fashion.

As said before, in due course of iterative process, it may happen that a food source cannot be improved after N number of trials and this ultimately leads to delay in optimization process or leads to poor convergence. To eliminate this, an exploration scheme has been incorporated in the algorithm via scout bee. 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 a Scout. Food source associated to the scout bee is abandoned and is initialized to random food source. In this version only on scout bee has been considered. This process is continued till the termination criterion is reached or required solution is obtained.

3 Cooperative Micro Artificial Bee Colony Algorithm

Cooperative Micro Artificial Bee Colony algorithm or simply CMABC involves concept of Cooperative Evolutionary Algorithms (CEAs), Micro Evolutionary Algorithms (Micro-EAs) and Artificial Bee Colony Algorithm. Following the pioneering work of Parsopoulos [3], these strategies when combined judiciously will produce

fruitful results. A micro-ABC will be discussed first and later cooperative methods link to this micro ABC resulting in CMABC.

Micro-ABC shares its similarities with its parent algorithm ABC. The only difference is the size of population, which is very small when compared to normal method. As food sources (solutions) has to be equally distributed among employed and on-lookers a population size of $NB=6$, can be of good choice [11]. Not only Micro-ABC any Micro-EA will have a hidden advantage and an explicit disadvantage (which has to be rectified). Due to less number of population, Micro-ABC is expected to converge rapidly to optimum (may or may not be global optimum). As mentioned in [3], the ratio of n/NB , is measure of hardness met by an algorithm for a problem considered. Literature studies and experimental results suggest that in many situations higher the ratio (>1) much harder the problem becomes for an algorithm. Hence the disadvantage of Micro-ABC is that it can be well suited for low-dimension problems.

This particular disadvantage can be alleviated by incorporating cooperative scheme in to Micro-ABC, which results in CMABC method. A divide and conquer approach has been used and it is represented by following set of equations. The original dimension of problem is decomposed in to small groups. Let n_1, n_2, \dots, n_k be K positive integers such that

$$n = \sum_{k=1}^K n_k$$

Here n corresponds to the dimension of problem considered. So, from the above equation it is clear that a solution vector of original problem can be divided into K sub components. Each subcomponent is addressed by different subpopulation, P_i of size NB_i , and dimension equal to $n_i, i=1, 2, 3 \dots K$. Hence, each subpopulation is given a task of minimizing its corresponding subcomponent, which has dimension very much less than the original problem dimension. Thus maintaining a good n/NB ratio.

As the subcomponent dimension differs from original dimension, one more hurdle met by algorithm is how to evaluate the individual with respect to an objective function? This problem can be approached by using a mechanism termed as information sharing, in which a buffer memory is created. All the subpopulations deposit their best individuals in this buffer memory. It is occasionally referred as *context vector* and it is defined as an *n-dimensional vector*, and is defined as $BM = (bm_1, bm_2, \dots, bm_n)$ where each subpopulation deposits its contribution in its respective k_{th} position. Suppose, if:

$$bs^{[k]} = (bs_1^{[k]}, bs_2^{[k]}, \dots, bs_{n_k}^{[k]})$$

is the n_k -dimension vector contributed by the k_{th} subpopulation, $P_k, k=1, 2, \dots, K$, then the *context vector* or buffer solution is given by

$$BM = [bs^{[1]}, bs^{[2]}, \dots, bs^{[K]}]$$

$$BM = \left[\underbrace{bs_1^{[1]}, \dots, bs_{n_1}^{[1]}}_{bs^{[1]} \text{ of } P_1}, \underbrace{bs_1^{[2]}, \dots, bs_{n_2}^{[2]}}_{bs^{[2]} \text{ of } P_2}, \dots, \underbrace{bs_1^{[K]}, \dots, bs_{n_k}^{[K]}}_{bs^{[K]} \text{ of } P_K} \right]$$

Hence, the i_{th} sub individual of the j_{th} subpopulation is defined as

$$x_i^{[j]} = \left(x_{i1}^{[j]}, x_{i2}^{[j]}, \dots, x_{i,n_j}^{[j]} \right)$$

This i_{th} individual is evaluated using the buffer vector, BM , by substituting the components associated to j_{th} subpopulation in the corresponding j_{th} position of buffer vector, where remaining components get unaffected.

$$f(x_i^{[j]}) = f(BM_i^{[j]}) \tag{5}$$

where,

$$BM_i^{[j]} = \left[bs_1^{[1]}, \dots, bs_{n_1}^{[1]}, \dots, \underbrace{x_{i1}^{[j]}, \dots, x_{i,n_2}^{[j]}}_{\text{individual } x_i^{[j]}}, \dots, bs_1^{[K]}, \dots, bs_{n_k}^{[K]} \right]$$

$$i = 1, 2, \dots, NB_j ; j = 1, 2, \dots, K$$

If $bs^{[k]} = x_g^{[k]}$ represents a best position then the global best solution will be the context vector of form

$$BM = \left[\underbrace{x_{g1}^{[1]}, \dots, x_{g,n_1}^{[1]}}_{x_g^{[1]} \text{ of } P_1}, \underbrace{x_{g1}^{[2]}, \dots, x_{g,n_2}^{[2]}}_{x_g^{[2]} \text{ of } P_2}, \dots, \underbrace{x_1^{[K]}, \dots, x_{g,n_k}^{[K]}}_{x_g^{[K]} \text{ of } P_K} \right]$$

The pseudo code of proposed method is presented below and an example value of subcomponent calculation is provided in section IV.

Pseudo code of CMABC

Input: K /* number of subpopulations*/ NB_i /* size of subpopulation*/
 n_i /* dimensions of subpopulation*/ $i=[1,2,\dots,K]$ /*
 M /* Context Vector*/

1. Initialize subpopulation randomly
2. Initiate context vector BM using above strategy
3. **While** (iterations!=maximum)
 For $k=1$ to K

For $i = 1$ to NB_k

1. Update individual $x_i^{[k]}$ with ABC Employed (onlookers) and scout bees. Evaluate $f(x_i^{[k]})$ and $f(BM)$
2. Apply Greedy mechanism for the above $x_i^{[k]}$ and pass components to context vector

End

End

End While

4. Record the solutions and print them
-

4 Experimental Section

As the proposed method derives its origin from a variant of DE a test suite similar to the DE was chosen so that comparison would be fare. A high-dimensional test suite was considered and the specifications are tabulated in Table 1. To highlight the advantages of CMABC we also compared it with normal ABC and also two cooperative variants of DE [3]. The parameters involved in this search process are recorded in Table 2.

Table 1. Benchmark Problems

| Function | Range |
|---------------------|-------------------|
| Sphere (F_1) | $[-100, 100]^n$ |
| Rosenbrok (F_2) | $[-30, 30]^n$ |
| Rastrigin (F_3) | $[-5.12, 5.12]^n$ |
| Griewank (F_4) | $[-600, 600]^n$ |
| Ackley (F_5) | $[-20, 20]^n$ |

Table 2. CMABC and ABC algorithm parameters

| Parameter | Description | Value |
|-----------------|---------------------------------|----------------|
| NB_k | subpopulation size [#] | 6 |
| FS_k | Food sources | NB_k |
| n_k | subpopulation dimension | 5 |
| max_iter | maximum iterations | 1000 |
| $limit_{CMABC}$ | limit parameter for ABC | $(NB_k/2)*n_k$ |
| $limit_{ABC}$ | limit parameter for ABC | 200 |

[#]equally distributed among employed and onlookers

As discussed earlier that ABC will follow one-to-one mapping between bee allotted and food source. Hence food source will be equal to half of total size of colony. Parameter *limit* plays key role in exploration point of view; a random value leads to poor performance of algorithm. Our previous studies state that if *limit* is set to product of employed bee and total dimension of problem, the results will be extremely good. Hence we applied similar strategy to CMABC as well, on other hand this concept will not come handy in ABC method (used for comparison in current study) as the iterations are less and the value of *limit* will be more than the iterations hence an optimum value of 200 is chosen [13]. The dimensions considered are in the multiples of 300 and individuals obtained are multiples of 360, hence ratios will be of integers and also aforementioned ratio will be less than 1 for both ABC and CMABC.

Table 3. Dimensions, Sobpopulations, Total Individuals

| Problem Dimension | Total number of individuals | Total number of subpopulations |
|-------------------|-----------------------------|--------------------------------|
| 300 | 360 | 60 |
| 600 | 720 | 120 |
| 900 | 1080 | 180 |
| 1200 | 1440 | 240 |

Example 1: Consider a problem having dimension 300, for this problem based on Table 2 parametric set up following calculations can be obtained.

- n or $D=300$; (dimension of problem)
- $K=n/n_k=300/5=60$; (total of K subcomponents)
- $Total\ Individuals=K*NB_k=60*6=360$;
- $No.\ of\ Employed\ (onlookers-CMABC)=NB/2=6/2=3$;
- $limit_{CMABC}=(NB_k*n_k)/2=(6*5)/2=15$;

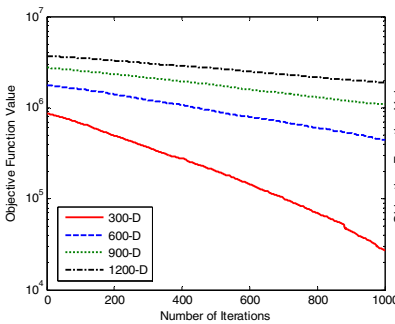


Fig. 1a Convergence of ABC w.r.t F_1

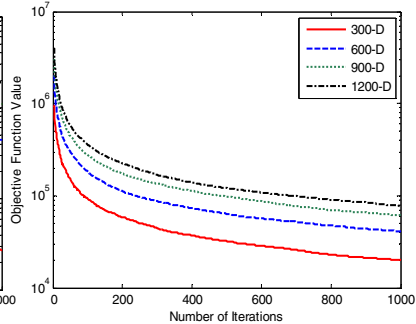


Fig. 1b Convergence of ABC w.r.t F_1

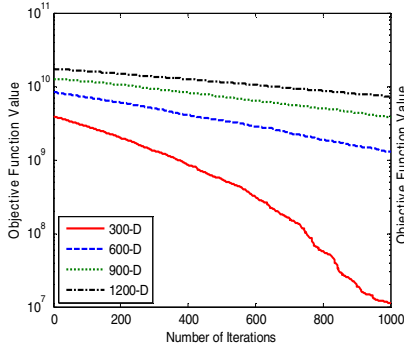


Fig. 2a Convergence of ABC w.r.t F_2

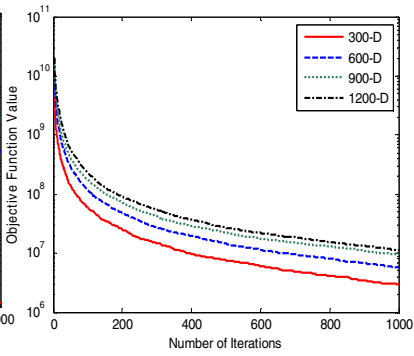


Fig. 2b Convergence of ABC w.r.t F_2

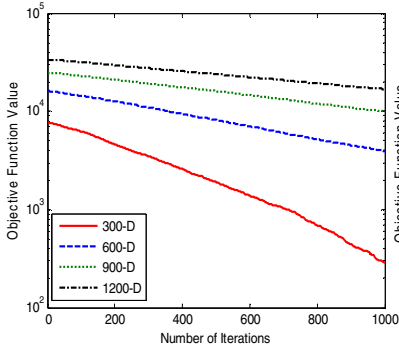


Fig. 3a Convergence of ABC w.r.t F_3

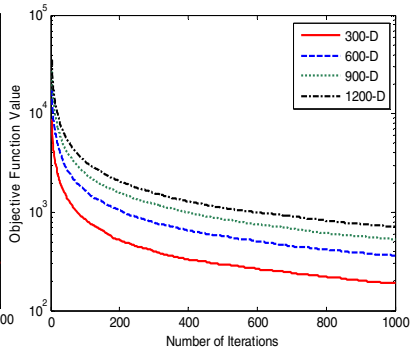


Fig. 3b Convergence of ABC w.r.t F_3

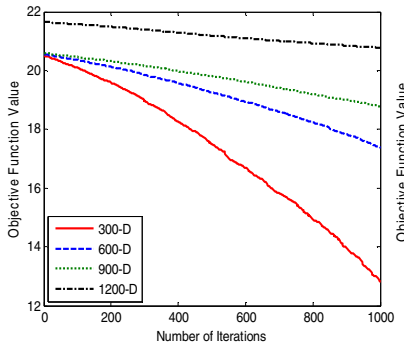


Fig. 4a Convergence of ABC w.r.t F_5

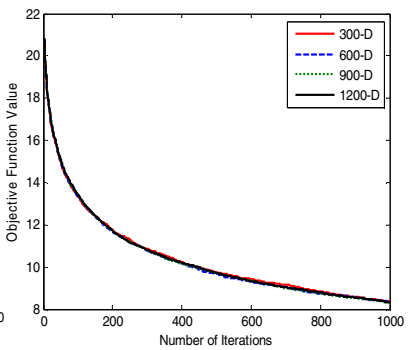


Fig. 4b Convergence of ABC w.r.t F_5

5 Discussions

Fig 1-4 provides the convergence plots for ABC method and CMABC methods (for various dimensions) respectively. For each and every instance ABC method faced a serious problem in converging towards optimum. From Fig 1(a)-4(a) it is also evident

that convergence deteriorated with increase in dimensionality, which is known as “curse of dimensionality” discussed in previous sections. More over for dimension above 1000 the convergence is very poor and is almost linear. On the other hand Fig 1(b)-4(b) depicts the convergence characteristics of CMABC method over dimensions ranging from 300-1200, from sphere function to Ackley function it is clear that convergence is not much affected with the increase in dimensions. This is mainly due to the added advantage of cooperative nature followed by micro nature of algorithm, which drives algorithm towards optimum more rapidly. Table 4-Table 8 provides the Mean (Std) results obtained for 25 independent test trails, for ABC and CMABC. To showcase the advantage of CMABC two cooperative micro variants of DE [3] are also compared. Tabular values reflect the superior performance of CMABC in all the cases considered

The only parameter that has to be chosen in the *CMABC* is the total size of population, where as for *COMDE* (variant of DE used for comparison) apart from size of population scaling factor F , followed by crossover rate CR , has to be chosen optimally. These two factors play a key role in DE and hence used should be careful in choosing values that balance harmonious relation. In *CMABC* *limit* is set as function of population and dimensionality and hence there will not be any difficulty in choosing optimal value. Also with help of *limit* parameter the subpopulation will have good exploration capability along with exploitation. For instance, a 300-D problem decomposed in to 60 subcomponents each having 5-subdimensions will have limit value equal to 15 ($NB/2 * n_k$). Hence each individual in a subpopulation will be subjected to a trail check for every 15 iterations and if any of them is not able provide better value; a scout mechanism is carried out.

These hidden advantages in proposed strategy cumulatively sum up to give highly promising results over the rest of them. The performance of proposed method can be further improved by providing adaptive nature for grouping of subcomponents.

Table 4. Sphere Function: Mean and Standard Deviation (std) values obtained for CMABC & comparative methods

| Dim(D) | ABC | CMABC | COMDE1 | COMDE2 |
|--------|----------------------------|--|----------------------------|----------------------------|
| 300 | 2.7302e+04 (9.5946e+03) | 1.9867e+04 (8.9724e+02) | 8.6876e+04 (1.5732e+04) | 9.3599e+04 (2.2051e+04) |
| 600 | 4.4255e+05 (1.9077e+04) | 4.0964e+04 (1.9525e+03) | 1.8909e+05 (2.0218e+04) | 2.0026e+05 (3.4124e+04) |
| 900 | 1.0821e+06 (2.4272e+04) | 6.1087e+04 (3.9624e+03) | 2.7960e+05 (2.6643e+04) | 3.1485e+05 (4.5966e+04) |
| 1200 | 1.8816e+06 (5.6735e+04) | 7.8102e+04 (1.6795e+03) | 3.5644e+05 (2.8171e+04) | 4.1765e+05 (3.6677e+04) |

Table 5. Rosenbrock Function: Mean and Standard Deviation (std) values obtained for CMABC & comparative methods

| Dim (D) | ABC | CMABC | COMDE1 | COMDE2 |
|---------|----------------------------|--|----------------------------|----------------------------|
| 300 | 1.1151e+07 (9.4697e+06) | 2.9605e+06 (2.0675e+05) | 1.2478e+08 (4.4738e+07) | 1.7087e+08 (3.9198e+07) |
| 600 | 1.2815e+09 (2.0128e+08) | 5.7833e+06 (7.1771e+05) | 2.5380e+08 (6.8593e+07) | 4.0705e+08 (6.5600e+07) |
| 900 | 3.7998e+09 (2.0632e+08) | 9.6210e+06 (6.5142e+05) | 3.7215e+08 (6.9986e+07) | 6.2499e+08 (9.1419e+07) |
| 1200 | 7.2564e+09 (2.3577e+08) | 1.1329e+07 (2.5777e+05) | 5.1075e+08 (8.6568e+07) | 8.7136e+08 (8.9369e+07) |

Table 6. Rastrigin Function: Mean and Standard Deviation (std) values obtained for CMABC & comparative methods

| Dim (D) | ABC | CMABC | COMDE1 | COMDE2 |
|---------|----------------------------|--|----------------------------|----------------------------|
| 300 | 1.0493e+03 (1.9893e+01) | 8.6733e+02 (9.4350e+00) | 1.6969e+03 (9.6112e+01) | 1.4579e+03 (1.1371e+02) |
| 600 | 4.1560e+03 (1.0854e+02) | 1.7383e+03 (9.6776e+01) | 3.3438e+03 (1.6715e+02) | 3.4638e+03 (1.2190e+02) |
| 900 | 8.3384e+03 (7.2774e+01) | 2.5386e+03 (3.7478e+01) | 5.0532e+03 (2.1358e+02) | 5.6513e+03 (1.7424e+02) |
| 1200 | 1.2702e+04 (2.4585e+02) | 3.4110e+03 (4.6170e+01) | 6.6904e+03 (1.6206e+02) | 7.8643e+03 (2.1178e+02) |

Table 7. Griewank Function: Mean and Standard Deviation (std) values obtained for CMABC & comparative methods

| Dim (D) | ABC | CMABC | COMDE1 | COMDE2 |
|---------|----------------------------|--|----------------------------|----------------------------|
| 300 | 2.8539e+02 (7.9797e+01) | 1.8944e+02 (1.5891e+01) | 8.2556e+02 (1.4660e+02) | 8.5437e+02 (1.8164e+02) |
| 600 | 3.9453e+03 (2.2046e+02) | 3.5998e+02 (2.7179e+00) | 1.6579e+03 (1.2883e+02) | 1.9105e+03 (2.7263e+02) |
| 900 | 9.9258e+03 (1.2836e+02) | 5.2569e+02 (2.2904e+01) | 2.4738e+03 (2.1885e+02) | 2.7540e+03 (2.9849e+02) |
| 1200 | 1.6818e+04 (2.7895e+02) | 7.1089e+02 (2.8326e+01) | 3.3135e+03 (2.4706e+02) | 3.7251e+03 (3.6052e+02) |

Table 8. Ackley Function: Mean and Standard Deviation (std) values obtained for CMABC & comparative methods

| Dim (D) | ABC | CMABC | COMDE1 | COMDE2 |
|---------|----------------------------|--|----------------------------|----------------------------|
| 300 | 1.2824e+01 (4.7000e-01) | 8.3703e+00 (1.5077e-01) | 1.3792e+01 (5.1525e-01) | 1.3949e+01 (6.2472e-01) |
| 600 | 1.7399e+01 (1.1987e-01) | 8.3690e+00 (1.1231e-01) | 1.3651e+01 (3.6977e-01) | 1.4605e+01 (3.6721e-01) |
| 900 | 1.8783e+01 (9.1087e-02) | 8.3266e+00 (7.0677e-02) | 1.3741e+01 (4.2847e-01) | 1.4902e+01 (3.2222e-01) |
| 1200 | 2.0779e+01 (7.5031e-02) | 8.3405e+00 (6.3217e-02) | 1.3794e+01 (3.0914e-01) | 1.5002e+01 (2.6469e-01) |

6 Conclusions

A cooperative frame work imbued with micro nature of Artificial Bee Colony is proposed and suggested for solving large scale optimization problems. The algorithm is made parameter free (trial and error) and the advantages of proposed method are validated on a benchmark suite. From the results obtained and comparisons with reported results it is very clear that CMABC proved to be very valuable tool for problems of high dimensions. As suggested by Li [5] random grouping methods may also be useful for these kinds of methods. Our future research will focus on strengthening the proposed approach for solving non-separable functions and also few real world intriguing problems as well. Exploitation schemes based on Cauchy and Gaussian mutations and their extensions to CAMC are also subject of interest.

References

1. Potter, M.A., De Jong, K.: Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation* 8(1), 1–29 (2000)
2. Potter, M., De Jong, K.: A cooperative coevolutionary approach for function optimization. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) *PPSN 1994*. LNCS, vol. 866, pp. 249–257. Springer, Heidelberg (1994)
3. Parsopoulos, K.E.: Cooperative micro-differential evolution for high-dimensional problems. In: *Proc: GECCO 2009*, pp. 531–538 (2009)
4. Vanden Bergh, F., Engelbrecht, A.P.: A cooperative approach to particle swarm optimization. *IEEE Trans. on Evol. Comput.* 8(3), 225–239 (2004)
5. Li, X., Yao, X.: Cooperatively Coevolving Particle Swarms for Large Scale Optimization. *IEEE Trans. on Evol. Comput.* 16(2), 210–224 (2012)
6. Yang, Z., Zhang, J., Tang, K., Yao, X., Sanderson, A.C.: An adaptive coevolutionary Differential Evolution algorithm for large-scale optimization. In: *IEEE Congress on Evol. Comput.*, pp. 102–109 (2009)
7. Zou, W., Zhu, Y., Chen, H., Sui, X.: A Clustering Approach Using Cooperative Artificial Bee Colony Algorithm. *Discrete Dynamics in Nature and Society* 2010, 16 pages (2010)
8. Zou, W., Zhu, Y., Chen, H., Zhu, Z.: Cooperative approaches to Artificial Bee Colony algorithm. In: *ICCAASM 2010*, vol. 9, pp. 22–24 (2010)
9. Koppen, M., Franke, K., Vicente-Garcia, R.: Tiny Gas for image processing applications. *IEEE Computational Intelligence Magazine* 1(2), 17–26 (2006)
10. Huang, T., Mohan, A.S.: Micro-particle swarm optimizer for solving high dimensional optimization problems. *Applied Mathematics and Computation* 181(2), 1148–1154 (2006)
11. Rajasekhar, A., Das, S., Das, S.: ABC: a micro artificial bee colony algorithm for large scale global optimization. In: *GECCO 2012*, pp. 1399–1400 (2012)
12. Karaboga, D., Basturk, B.: A Powerful and Efficient Algorithm for Numerical Optimization: Artificial Bee Colony (ABC) algorithm. *J. of Global Optim.* 3(39), 159–172 (2007)
13. Akay, B., Karaboga, K.: A modified Artificial Bee Colony algorithm for real-parameter optimization. *Information Sciences* 192, 120–142 (2012)
14. Rajasekhar, A., Abraham, A., Pant, M.: Levy mutated Artificial Bee Colony algorithm for global optimization. In: *2011 IEEE Conf. on Systems, Man and Cybernetics*, pp. 655–662 (2011)
15. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review*, 1–37 (2012)

Improvement in Genetic Algorithm with Genetic Operator Combination (GOC) and Immigrant Strategies for Multicast Routing in Ad Hoc Networks

P. Karthikeyan¹ and Subramanian Baskar²

¹Department of Information Technology

²Department of Electrical and Electronics Engineering

Thiagarajar College of Engineering, Madurai 625015, Tamilnadu, India

{karthikit,sbeee} @tce.edu

Abstract. In this paper, an improved Genetic Algorithm (GA) is proposed for solving multicast routing problem by optimizing combined objectives of network lifetime and delay. This algorithm employs Genetic Operator combination (GOC) and immigrant strategies. The GOC contains modified topology crossover, node and energy mutations. Immigrant strategies are the specific replacement operators designed for dynamic optimization problems and it is naturally suited for multicast routing in ad hoc networks. The random immigrant with random replacement, random immigrant with worst replacement, elitism based immigrant and hybrid immigrant strategies are combined with GOC individually, and formed four different algorithms. The performance of these algorithms is evaluated in different size networks through simulation. The results of the proposed algorithms are compared with other existing algorithms using nonparametric statistical tests with average ranking. These test results endorse that the proposed algorithms improve the performance of GA in solving multicast routing problems effectively.

Keywords : Ad Hoc Networks, Multicast Routing, Genetic Operator Combinations (GOCs), Immigrant Strategies.

1 Introduction

Ad hoc networks are decentralized infrastructure less networks. The topology of this network is changes depending on the nodes mobility. The communication between two nodes is carried out either directly or with the help of intermediate nodes which belong to the same network and is influenced by the nodes transmission range. This network facilitates numerous applications like video conferencing, military, vehicular network, content distribution and aquatic applications [1]. Ad hoc networks are characterized by non-restricted mobility and easy deployment, which make them very popular [2]. Multicasting [3, 4] is one type of service in ad hoc networks where the source node sends the message to many sinks through more than one path. The life of the ad hoc network depends on the individual node's residual battery. If the battery

power consumption is high in any one of the nodes, the chances of network's life reduction due to path-breaks are also more. This may lead to packet loss during packet forwarding in the multicast network [5]. The objectives considered in multicast routing problem include minimization of routing cost [6], minimization of link delay, maximization of the network lifetime, and maximization of the throughput [7]. The constraints in multicast routing problem are loop formation [8], link breaks [9], fixed bandwidth, fixed end-to-end delay and fixed link capacity [10]. Multicast routing is a Non-deterministic Polynomial (NP) hard problem for large scale and wide area network [11].

Many researchers have applied Genetic Algorithm (GA) for solving multicast routing with various coding and genetic operators [12-17]. The performance comparison of ST and Prufer coding for various crossover and mutation methods are discussed in [13-16]. The performance of GA is also depends on genetic operators used for the particular coding adopted. Karthikeyan et al. have proposed different genetic operators such as Modified Topology Crossover (MTC), Energy-II and Energy-III mutations and Genetic Operator Combinations (GOCs) for improving the performance of GA for multicast routing problems. Recently, to improve the performance of GA, different immigrant strategies have employed for handling Multicast routing and similar optimization problems. Tinos et al. have proposed Random Immigrants Genetic Algorithm (RIGA) for this problem. Yang et al. have proposed Elitism based Immigrant Genetic Algorithm (EIGA) to overcome the drawback of the RIGA in a slowly changing environment.

In this paper, an improved Genetic Algorithm (GA) with GOC and immigrant strategies is proposed for solving multicast routing problem. The immigrant strategies such as Random Immigrant (RI) [18], Elitism based Immigrant (EI) and Hybrid Immigrant (HI) [19] are considered in the proposed framework. These immigrant strategies are combined with GOC and formed several algorithms. Simulations are carried over in different networks. The obtained results of the proposed algorithms are compared with existing algorithms using non parametric statistical tests such as Friedman, Aligned Friedman and Quade [20].

The remainder of this paper is organized as follows: Section 2 discusses the problem description. Section 3 explains the proposed genetic algorithm. Section 4 elaborates the simulation results and finally Section 5 contains the conclusions and future work.

2 Problem Description

Assume that a network is represented as a graph $G = (N,L)$, where N and L denote the set of nodes and links, respectively. The sample weighted graph of ad hoc network model is given in Fig. 1. The nodes are labeled as 1, 2, 3... n.

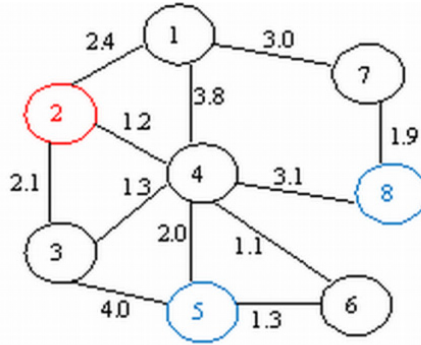


Fig. 1. Graph

Let $s \in N$ be the source and $M \subseteq \{N - s\}$ be the set of destinations. In a multicast tree $T(s,M)$, the total number of nodes is denoted as $|N|$, the total number of links is denoted as $|L|$. The multicast tree $T(s,M)$ can be generated from the nodes of graph G through which the source will send the message to multiple destinations. Multicast routing is a challenging service of ad hoc networks. Based on the paths and the routing parameter, a multicast tree is generated from the graph. In this work, lifetime of the multicast tree and the delay of each link are considered as routing metric. The delay function is defined as $Delay(l): L \rightarrow R^+$. R^+ is the set of positive real numbers. The total delay of the network is given in Eq. (1).

$$Delay(T(s, M)) = \sum_{l \in T(s, M)} Delay(l) \tag{1}$$

(i, j) denotes a link with node i and j which can transmit a packet to each other with a transmission power less than the maximum transmission power at each node, where $l(i, j) \in L$. The set of nodes connected to node i by links is denoted as Deg_i . Each node i is assumed to have Residual Battery Energy RBE_i . Let φ_{ij} denote the flow (in datagram units/s) from node i to node j . Assume that the energy consumption of node i to transmit a unit of datagram to node j is Λ_{ij} . Based on the assumption that φ_{ij} and Λ_{ij} values for all the nodes are same, the datagram flow rate C_{ij} is equated to the product of φ_{ij} and Λ_{ij} . i.e. $C_{ij} = \varphi_{ij} * \Lambda_{ij}$. The lifetime of node i is given in Eq. (2).

$$t_i = \frac{RBE_i}{\sum_{j \in Deg_i} C_{ij}} \tag{2}$$

The total network life time of the network is given in Eq. (3).

$$t_{net} = \min_{i \in N} \{t_i\} \tag{3}$$

In Fig. 1, node 2 is considered as a source node, and nodes 5 and 8 are designated as destination nodes. There are many paths to reach the destination nodes from the source node. Sample paths include [2-4-8, 2-3-5], [2-1-4-8, 2-3-4-5] etc. This can be obtained

by forming spanning trees with source node as the root and destination node as one of the leaf nodes as given in Fig. 2. L_i is the location index given near each node.

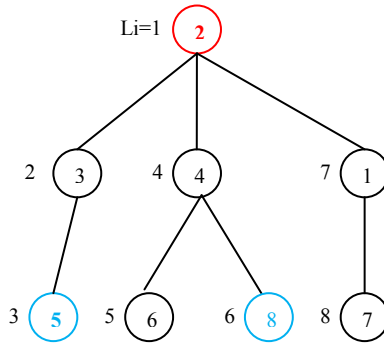


Fig. 2. Spanning Tree for the graph in Fig.1

From the tree, the paths to reach the destinations are identified. Then delay and network life time are calculated for evaluating the set of spanning trees. When calculating the path from the source to multiple destinations, battery power of the source node is not taken into account. It is assumed that the source node contains unlimited power. The node locations are determined using Global Positioning Systems (GPS). The nodes moved infrequently and hence the GPS has to operate for a very less number of periods and worked in regular time intervals. This is for determining the locations of the network nodes based on which the distance between the two nodes is calculated. The distance metric is directly proportional to the overall delay of the links.

In this paper, multiple objectives such as maximization of the network lifetime and the minimization of the time delay are considered. However, the objectives are combined into a single maximization objective [14] as given in Eq. (4).

$$OF(T(s, M)) = \alpha * F1 * F2 \tag{4}$$

Where,

α - Positive real coefficient

$OF(X)$ - Objective value for the individual X

$F1$ - $\frac{1}{Delay(T(s, M))}$

$F2$ - t_{net}

The number of broken links is taken as a constraint. During the generation of spanning trees for the given graph, some of the trees may have broken links. Trees with broken links are not valid. To handle this, a penalty parameter-less constraint handling scheme is employed [21].

3 Proposed Genetic Algorithm

Genetic Algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. GA belong to the larger class of Evolutionary Algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover [22]. Currently GA stands up as a powerful tool for solving complex, nonlinear, non-smooth, discrete optimization problems.

In a GA, a population of strings (called chromosomes), which encodes candidate solutions to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings such as integer, real, mixed integer are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population based on their fitness, and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. Coding, GOC and immigrant strategies used in the proposed GA for multicasting routing are discussed in the following sections.

3.1 Sequence and Topology (ST) Coding

The integer coding scheme is employed to represent nodes and location index of the parent node. In this coding scheme, a unique identification number is assigned using a preorder-visited method which is from top to bottom, and then from left to right in order to visit a multicast tree. The location index value is started from 1 to the number of nodes present in the tree. There are two parts in the chromosome. The first part is the sequence part. The node identification numbers are continuously inserted until the last node is reached. The second part of the chromosome is the topological part in which the parent's location index values of all the nodes are stored. The length of the sequence and topological parts are equal to the number of nodes present in the network. The first element of the topological part is always 0 because the parent node index value of the route node is 0. For example, the spanning tree given in the Fig. 2 is converted by ST encoding and it is given in Fig. 3. The sequence part is created based on the location index order. Topological part is formed by mentioning the parent index of each gene present in the sequential part. For example, the third gene in the sequence part is 5 and its parent (i.e. node 3) index value is 2 which is placed in third gene of topological part.

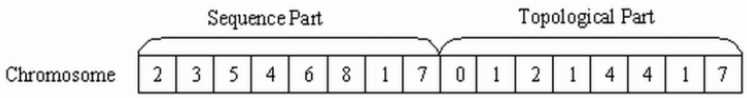


Fig. 3. ST coding illustration

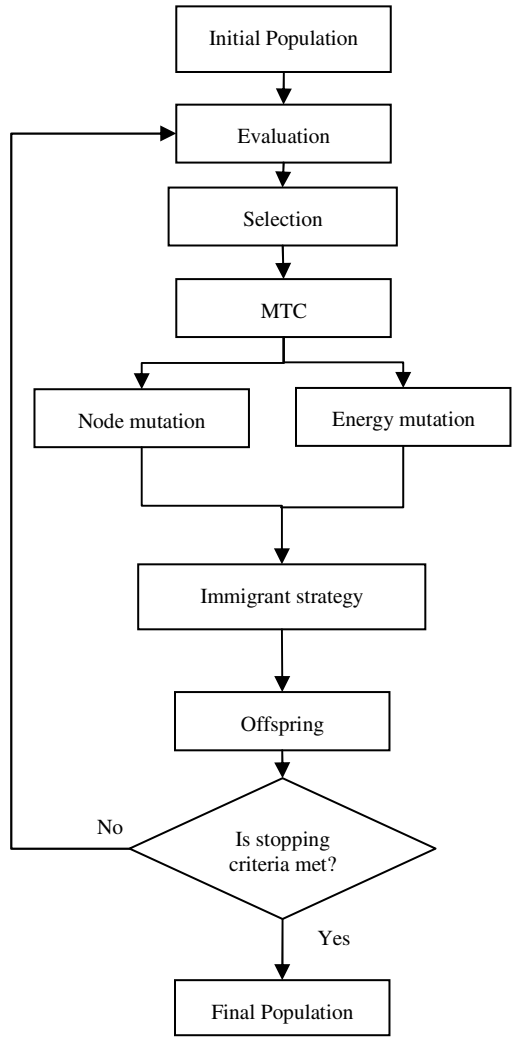


Fig. 4. Proposed GA

3.2 Genetic Operator Combination (GOC) and Immigrant Strategies

In general, the performance of the GA is highly dependent on the genetic operators used and their parameter settings. Also, during different evolution stages, different genetic operators are found to be more effective than others. Hence, in this paper GOC1 [12] is used due to its better performance than the other operators. The GOC1 includes modified topology crossover (MTC), node mutation and energy mutation operators.

Recently, immigrant strategies are introduced for improving the performance of GA. The GA performance may degrade due to network topology change. In such a dynamic environment, immigrant schemes are helping the GA to converge into the right solution space. The immigrant schemes considered for this multicast routing problem are Random Immigrant with Random Replacement (RI-RR), RI with Worst Replacement (RI-WR) [18], Elitism based Immigrant (EI) and Hybrid Immigrant (HI) [19]. RI-RR is generating a chromosome randomly and replaced randomly in the population where as the RI-WR is replaced with the worst individual. The EI is replaced best chromosome in each generation with worst one. HI is a combination of both RI-RR and EI. The proposed GA is illustrated in Fig.4.

3.3 Initial Population

The initial population is generated using uniform random numbers. The generated trees fitness values are evaluated. If the chromosome is invalid, then the numbers of broken links are calculated and the same is assigned to the constraint violation for that chromosome.

3.4 Penalty Parameter-Less Constraint Handling Scheme

In this paper, a penalty parameter-less constraint handling scheme proposed by Deb is used to handle the number of broken links. To handle this constraint, the Eq. (7) is used.

$$fitness_i(X) = \begin{cases} OF_i(X), & \text{if } \varphi(X) = 0 \\ OF_{worst} - \varphi(X), & \text{Otherwise} \end{cases} \quad (5)$$

Where,

$OF_i(X)$ - Objective function value of the i^{th} solution of chromosome X

OF_{worst} - Objective function value of the worst solution

$\varphi(X)$ - Equality constraint

Hence, without using a penalty factor, fitness function is determined by combining objective and constraint violations.

4 Simulation Results

The performance of GA with immigrant strategies is tested on four different ad hoc networks with varying nodes such as 10, 20, 30 and 40. Waxman's random graph generator [23] is used for generating the network graph randomly. The link weights

are calculated based on the nodes distance and it is assumed as the delay values in milliseconds. All simulations run on a Core 2 Duo Intel processor operating with 1GB RAM which is connected with the high end multiuser server with MatLab. To fix the suitable GA parameters such as population size (pop-size), mutation probability (P_m), and crossover probability (P_c) initial trial simulations are conducted with different combinations of parameters before taking actual results. GA parameters are fixed at pop-size=100, mutation probability $P_m = 0.3$ and crossover probability $P_c = 0.7$. Being a stochastic optimization algorithm, to get meaningful conclusions, statistical performance measures are calculated with 20 independent runs. Statistical tests such as Friedman, Aligned Friedman and Quad are used to determine the best algorithm through average ranking. The significance of the first rank method is addressed by the p value obtained from the tests such as Holm, Hommel and Holland. Initially a pool of genetic algorithms is created with various genetic operators such as crossover, mutations and immigrant strategies. Table 1 is illustrates variants of genetic algorithms namely GA1 to GA4 with corresponding immigrant strategies and its probability values.

Table 1. Genetic Algorithms with corresponding immigrant strategies and its probability

| Algorithm | Immigrant strategy | Probability |
|-----------|--------------------|-------------|
| GA1 | RI-RR | 0.3 |
| GA2 | RI-WR | 0.5 |
| GA3 | EI | 0.5 |
| GA4 | HI | 0.3 |

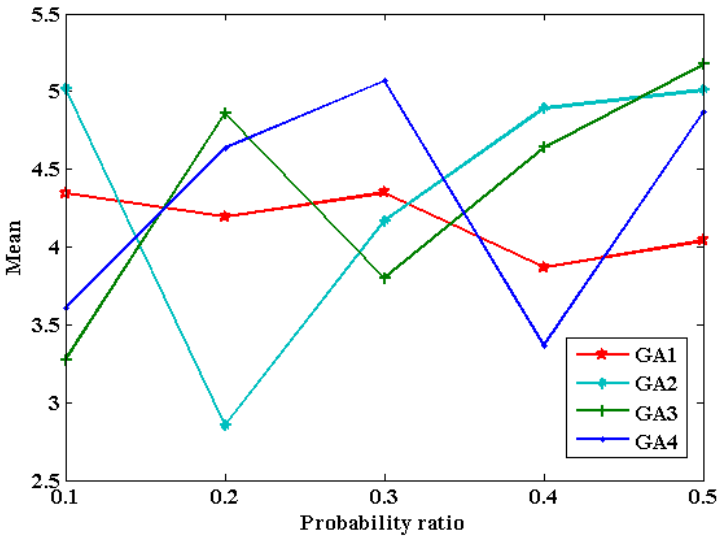


Fig. 5. Mean of the algorithms with different probability

The MTC, node mutation, and energy mutation are used in GA variants as these operators gave better results [12]. Different immigrant strategies are added along with that to form the various genetic algorithms. The immigrant probabilities of different GA may vary. Due to their better performance, they are considered. This is shown in Fig.5.

With the mean obtained from 20 runs from the proposed GA1 to GA4 and existing algorithms of [12, 14, and 15] in different networks, the average rankings are generated in Friedman, Aligned Friedman and Quade tests. This is given in Table 2.

Table 2. Average rankings of the algorithms

| Algorithms | Friedman | Aligned Friedman | Quade |
|------------|------------|------------------|-------------|
| GA1 | 3.72 | 75.625 | 3.79 |
| GA2 | 3.7 | 60.349 | 3.49 |
| GA3 | 3.92 | 70.025 | 4.16 |
| GA4 | 3.85 | 64.35 | 3.82 |
| GOC1 | 4.22 | 76.075 | 4.21 |
| GOC2 | 4.52 | 76.025 | 4.17 |
| GAST | 5.9 | 112.6 | 6.12 |
| ST | 6.14 | 108.949 | 6.23 |

The average rankings show that the proposed GA2 is performing well when compared with other algorithms. This is highlighted in Table 2. It is also necessary to confirm that GA2 is how significantly better than the other algorithms. This is done by obtaining p value with the level of significance $\alpha = 0.05$ in Holm, Hommel and Holland tests. These values are listed in Table 3.

The Holm and Hommel tests reject the hypothesis that have a p value ≤ 0.01 . The Holland test rejects the hypothesis that have a p value ≤ 0.0102 . From the results of Table 3, it is clear that the existing algorithms GOC2, GAST and ST are rejected and it implies that the proposed GA2 is significantly better than these algorithms. At the same time, GA1, GA3, GA4 and GOC1 have not been rejected. Even though they were behind in average ranking with GA2, the performance of these algorithms are also better. They have been highlighted in the Table 3. With these test results, it is proved that the proposed GA2 improves the performance of GA for multicast routing in ad hoc networks.

Table 3. The p value with level of significance $\alpha = 0.05$

| Algorithms | Friedman/Aligned Friedman/ Quade | | |
|-------------|----------------------------------|----------------|----------------|
| | Holm | Hommel | Holland |
| GA1 | 0.05 | 0.05 | 0.05 |
| GA3 | 0.01667 | 0.01667 | 0.01695 |
| GA4 | 0.025 | 0.025 | 0.0253 |
| GOC1 | 0.0125 | 0.0125 | 0.01274 |
| GOC2 | 0.01 | 0.01 | 0.0102 |
| GAST | 0.00833 | 0.00833 | 0.00851 |
| ST | 0.00714 | 0.00714 | 0.0073 |

5 Conclusions and Future Work

In this paper, for improving the performance of GA, GOC and immigrant strategies are proposed to address multicast routing problem in ad hoc networks. The four immigrant strategies used are random immigrants with random replacement, random immigrants with worst replacement, elitism based immigrant and hybrid immigrant. An augmented objective function is formulated by combining network lifetime and time delay. Simulations are carried over in different genetic algorithms with GOC and various immigrant strategies on different size networks. The results obtained from simulations are experienced through statistical tests. The test results clearly show that the proposed genetic algorithms such as GA1 to GA4 are better than the existing algorithms for solving multicast routing problem in ad hoc network.

In the proposed GAs, the genetic operator combinations and immigrant strategy are working in fixed probability in each algorithm. As a part of future work, it is decided to keep all the immigrant strategies with an adaptive probability in a single genetic algorithm with an ensemble manner for producing better results. Since it is an ad hoc network, it is also plan to test on different topologies of various networks.

References

1. Kong, J.: Building underwater ad-hoc networks and sensor networks for large scale real-time aquatic applications. In: *Int. Conf. Milit. Comm.*, Atlantic City, NJ (2005)
2. Sesay, S., Yang, Z., He, J.: A survey on mobile ad hoc wireless network. *InfoTech*, 168–175 (2004)
3. Oliveira, C., Pardalos, P.: A survey of combinatorial optimization problems in multicast routing. *Comp. and Oper. Res.* 32, 1953–1981 (2005)
4. Wang, B., Hou, J.: A survey on multicast routing and its QoS extensions: problems, algorithms, and protocols. *IEEE Trans. Net.* 14, 22–36 (2000)
5. Wang, B., Gupta, S.K.S.: On maximizing lifetime of multicast trees in wireless ad hoc networks. In: *Intl. Conf. Para. Proc.*, Kaohsiung, Taiwan (2003)
6. Haghghat, A.T., Faez, K., Dehghan, M.: GA-based heuristic algorithms for QoS based multicast routing. *Know Bas. Sys.* 16, 305–312 (2003)
7. Baumann, R., Heimlicher, S., Strasser, M., Weibel, A.: A survey on routing metrics. TIK Report, Computer Engineering and Networks Laboratory, ETH-Zentrum, Switzerland (2007)
8. Sateesh Kumar, P., Ramachandram, S.: Genetic zone routing protocol. *Theo. and Appl. Info. Tech.* 4, 789–794 (2008)
9. Eiben, A.E., Smith, J.E.: *Introduction to Evolutionary computing Genetic Algorithms*. Springer (2010)
10. Sun, B., Pi, S., Gui, C., Zeng, Y., Yan, B., Wang, W., Qin, Q.: Multiple constraints QoS multicast routing optimization algorithm in MANET based on GA. *Progress in Nat. Sci.* 18, 331–336 (2008)
11. Koyama, A., Nishie, T., Arai, J., Barolli, L.: A GA-based QoS multicast routing algorithm for large-scale networks. *Int. J. of High. Perf. Comp. & Net.* 5, 381–387 (2008)

12. Karthikeyan, P., Baskar, S., Alphones, A.: Improved genetic algorithm using different genetic operator combinations (GOCs) for multicast routing in ad hoc networks. *Soft. Comput.* (2012), doi:10.1007/s00500-012-0976-4
13. Yen, Y.S., Chao, H.C., Chang, R.S., Vasilakos, A.: Flooding-limited and multi-constrained QoS multicast routing based on the genetic algorithm for MANETs. *Math. and Comp. Mode* 53, 2238–2250 (2011)
14. Yen, Y.S., Chan, Y.K., Chao, H.C., Park, J.H.: A genetic algorithm for energy-efficient based multicast routing on MANETs. *Comp. Comm.* 31, 2632–2641 (2008)
15. Cao, Q., Zhou, J., Li, C., Huang, R.: A genetic algorithm based on extended sequence and topology encoding for the multicast protocol in two-tiered WSN. *Exp. Sys. with Appl.* 37, 1684–1695 (2010)
16. Chiang, T.C., Liu, C.H., Huang, Y.M.: A near-optimal multicast scheme for mobile ad hoc networks using a hybrid genetic algorithm. *Exp. Sys. with Appl.* 33, 734–742 (2007)
17. Jain, S., Sharma, J.D.: QoS constraints multicast routing for residual bandwidth optimization using evolutionary algorithm. *Int. J. Comp. Theo. and Eng.* 3, 211–216 (2011)
18. Tinos, R., Yang, S.: A self-organizing random immigrants genetic algorithm for dynamic optimization problems. *Genet. Program. Evol. Mach.* 8(3), 255–286 (2007)
19. Yang, S., Tinos, R.: A hybrid immigrants scheme for genetic algorithms in dynamic environments. *Int. J. Automat. Comput.* 4(3), 243–254 (2007)
20. Derrac, J., Garcia, 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 Evol. Comp.* 1, 3–18 (2011)
21. Deb, K.: An efficient constraint-handling method for genetic algorithms. *Comp. Meth. in Appl. Mech. and Eng.* 186, 311–338 (2000)
22. Goldberg, D.E.: *Genetic algorithms in search, optimization, and machine learning.* Addison Wesley, Boston (1989)
23. Salama, H.F., Reeves, D.S., Viniotis, Y.: Evaluation of multicast routing algorithms for real-time communication on high-speed networks. *IEEE J. on Sel. Ar. in Comm.* 15, 332–345 (1997)

Ensemble of Dying Strategies Based Multi-objective Genetic Algorithm

Rahila Patel¹, M.M. Raghuwanshi², and L.G. Malik³

¹ Research scholar, G.H. Rasoni College of Engineering, Nagpur

² Rajiv Gandhi College of Engg & Research, Nagpur

³ G.H. Rasoni College of Engineering, Nagpur

rahila.patel@gmail.com

Abstract. Excessive dying in nature causes reduction in diversity and leads to extinction of organism. In this work to avoid excessive dying we propose explicit dying strategies as part of Genetic Algorithm. A solution is removed from next generation population in a deterministic way by using dying strategy. Multi-objective Genetic Algorithm (MOGA) takes decision of removal of solution, based on one of these three strategies. Experiments were performed to show impact of dying of solutions and dying strategies on the performance of MOGA. Further to improve performance of MOGA an ensemble of dying strategy is proposed. Ensemble of Dying Strategy based MOGA (EDS-MOGA) has been implemented and its results show that ensemble of dying strategy has given better performance than MOGA with single dying strategy.

1 Introduction

Genetic Algorithm (GA) mimics the process of natural selection and is robust tool for search and optimization problems [1]. Many researchers have proven this. Review done by suganthan et.al in [14] present the development of multi-objective evolutionary algorithms (MOEAs) primarily during the last eight years. The survey covers algorithmic frameworks such as decomposition-based MOEAs (MOEA/Ds), memetic MOEAs, co-evolutionary MOEAs, selection and offspring reproduction operators, MOEAs with specific search methods, MOEAs for multimodal problems, constraint handling and MOEAs, computationally expensive multi-objective optimization problems (MOPs), dynamic MOPs, noisy MOPs, combinatorial and discrete MOPs, benchmark problems, performance indicators, and applications. In addition, some future research issues are also presented.

Genetic algorithm uses selection, crossover and mutation operators to evolve a set of solutions of current generation. Selection operator is then used to select fit solutions from current generation as next generation population [1]. Weaker Solutions which are not selected die out and do not reappear. Thus dying is implicit part of selection mechanism. Solutions having better fitness produce fitter offspring and selection strategies are likely to select both parent and their offspring. Solutions (parent solution) present in first generation inherit their properties to offspring solutions by

using cross over operator. In subsequent generations good properties of parent solutions are carrying forward by offspring solutions generated by crossover operator. Problem with this selection strategy is that, after few generation whole population is dominated by presence of few solutions from initial population and their offspring i.e. trail of very few solutions from initial population reach to final generation and most of the solutions die out somewhere in between generations. Thus selection strategy based on survival of fittest reduces diversity and convergence of solutions. So, a new mechanism is needed to avoid excessive dying of solutions.

In this work to control excessive dying of solutions, dying has been made explicit part of evolutionary process and three strategies for dying of solution have been proposed and implemented. A solution is deterministically removed from next generation population by using one of the three proposed dying strategies. Impact of dying rate of solutions on the performance of GA has been studied. Idea of gradual dying is modeled and a new framework of MOGA has been used to demonstrate the same.

According to No Free Lunch (NFL) Theorems for optimization if an algorithm performs well on one set of problems then it will perform poorly on all others [10]. With this observation we propose ensemble of dying strategy based MOGA (EDS-MOGA) that can optimize given set of problems.

The remainder of the paper is organized as follows. Section 2 briefs about proposed dying strategies. In Section 3, experimentation and results of three strategies are given. Section 4 presents ensemble of dying strategy based MOGA and its experimental results. Section 5 compares performance of EDS-MOGA with individual strategy of dying and NSGA-II-MPX. Finally section 6 with draws conclusions.

2 Three Dying Strategies

A thought; dying of parent solution; opposite of selection of parent solution, is materialized in this work. Proposed three strategies of dying or parent removal (*ParRem*) are given below.

ParRem1: Remove solution having minimum distance from one or more solutions of next generation population. In this strategy before removing a solution from next generation population distance between all the solutions is checked. Since similarity means uniformity and dissimilarity means diversity, according to this strategy similar solution should die out and dissimilar solution should remain in population in order to have diversity in population. The solution having minimum distance from its neighbor solution will be removed from next generation population. Distance between solutions is calculated in objective space and distance measure used is Euclidean distance. Pseudo code for distance calculation is given below.

```
Initialize min_d = 9999,
for j = 1 : N-1
  for i = j+1 : N
    d = distance between first parent( $i^{th}$ ) and second parent ( $j^{th}$ ) parent
    if (d < min_d)
      min_d = d;
```

```

        idx = i;
    end if
endfor
endifor

```

where N is population size, min_d is minimum distance between two parent solutions. idx is index of parent having minimum distance. A parent having idx index will be removed from next generation population.

ParRem2: Remove solution having maximum SNOV (Summation of Normalized Objective Value). In this strategy to make effective range of all the objective functions equal objective values are normalized. After normalization effective range of all the objective function will be zero to one. Assign SNOV to each solution in next generation population. The SNOV will be treated as single scalar fitness of the solution. Now remove solution having highest SNOV value (for minimization of objective). Pseudo code for SNOV calculation is given below.

```

for m = 1 : M (number of objectives)

```

Find the maximum and minimum objective values of the m^{th} objective and calculate the range of the m^{th} objective.

Normalize the m^{th} objective values of all members using the equation:

$$f_m(x) = \frac{f_m(x) - f_{\min}}{f_{\max} - f_{\min}}$$

where f_m is the normalized m^{th} objective value.

```

end for

```

```

for i = 1 : N

```

Sum all normalized objective values of the member to obtain a single value.

```

endfor

```

ParRem3: Remove solution having poor fertility count. Frequency of reproduction is an important parameter in determining species survival: an organism that dies young but leaves numerous offspring displays, according to Darwinian criteria, much greater fitness than a long-lived organism leaving only one [11][12].

In this strategy, parameter `fertility_count` keep record of frequency of reproduction of a parent solution. Initially when population is initialized, zero `fertility_count` is assigned to all the solution in the population. If a parent solution produces offspring better than the parent solution then `fertility_count` of parent solution is incremented by one. Now `fertility_count` of parent solution is assigned to better offspring solution. In next generation population, dying of a solution is insured on the basis of `fertility_count` of the solution. A solution having minimum `fertility_count` will be removed from next generation population. Pseudo code is given below.

```

for i = 1 to N

```

`fertility_count(i) = 0;`

```

endfor

```

```

for gen = 1 to maxgen

```

```

    for i = 1 to N
    If offspring solution wins the multi-level tournament then fertility_count of first parent solution is incremented by 1 and assigned as fertility_count to winner offspring solution.
    endfor
    Remove solution having lowest fertility_count
    ...
endfor

```

3 Experimentation and Results

A Multi-objective Genetic Algorithm with Dying strategy has been coded using Mat-Lab 7.1. Experimental study is performed to investigate the behavior of the proposed three strategies of Dying in terms of convergence and diversity. The MOGA procedure used for experimentation is given below.

Algorithm

Begin:

Initialization:

Set the parameter values for maxgen (Maximum number of generation), Dying rate in % (Perc), Size of population (N), Probability distribution index for MPX [3] operator, number of parents and number offspring generated by MPX operator.

Processing:

1. Generate N solutions randomly then tune them using OBL and keep tuned solutions in *POP*
 2. Assign number 1,2,...,N as parent number sequentially to each solution in *POP*
 3. Calculate $Gen_Rem = (maxgen / (N * Perc))$
 4. $iGen_Rem = Gen_Rem$
 5. for gen = 1 : maxgen
 - Set *off_pool* and *Next_gen* to be the empty set
 - for i = 1:N
 - o Select first solution sequentially and second solution randomly from *POP*
 - o Now apply MPX crossover operator on the selected solutions and generate two offspring. Assign parent number of first parent to both the offspring.
 - o Using Multi-level tournament selection select best among first parent and offspring solutions
 - o Copy best solution in *Next_gen* and put the offspring which is not selected in the *off_pool*
- endfor
- if gen == Gen_Rem
- o Use dying strategy to remove one solution from *Next_gen* and insert one solution, from *off_pool*, whose parent number is not same as the removed solution.

```

    o Gen_Rem = Gen_Rem + iGen_Rem
    o POP = Next_gen
else
    POP = Next_gen
end ifelse
end for
End

```

Initialization and Tuning the Population: In proposed algorithm population tuning is done by using opposition based learning [5]. A solution is sequentially selected from initial population and its opposite solution is calculated by using opposition based learning as given in [13]. Now tournament is played between solution and opposite solution and winner is selected as tuned solution and loser is discarded. This procedure is repeated for every solution. If there are N solutions in the initial population then N tune solutions are generated.

Tournament Selection: In this algorithm multi-level tournament selection is used. In first level of tournament selection, tournament is played between two offspring solutions. In second level, tournament is played between winner offspring solution and First parent. The winner of second level of tournament is selected as best solution and becomes member of next generation population.

MPX (Multi-parent Crossover with Polynomial distribution) operator is used for crossover. One can find details of MPX operator in [3]. 30 independent runs have been taken for each problem with different strategies. Dying rate is chosen empirically. Experimental parameter settings used are:

- Population size (N) = 300
- Maximum no. of generation (maxgen) = 3000
- Dying rate (Perc) = 10%, 30%, 40%, 50%, 60%, 90%
- Probability Distribution index (μ) for MPX crossover operator = 1
- Number of Parents = 2
- Number of children = 2

Test functions: The test problems (UF1-UF9) in this work are taken from IEEE CEC2009 special session and competition [4]. Functions UF1-UF7 has two objectives and UF8 & UF9 are three objective functions.

Metric used for Performance measure:

- Inverted Generational Distance (IGD): The IGD (Inverted Generational Distance) metric is used as performance indicator to quantify the quality of the obtained results. The IGD metric measures “how well is the Pareto-optimal front represented by the obtained solution set”. [4]
- SPREAD (Δ): This metric is used to measure diversity among solution. This metric takes care of uniform distribution and extent of distribution of obtained solution. [2]

3.1 Results

We have proposed three strategies of dying of solution, implemented them and seen their performance on function UF1-UF9. Table 1 shows best performance (MeanIGD and SPREAD) of three strategies of dying of solution on function UF1-UF9. Also Table 2 shows Statistical Sum of Problems for which each strategy obtains significantly better results. During experimentation it is observed that for all three strategies best results for MeanIGD and SPREAD have been obtained for dying rate at or below 50%. Hence we can say that excessive dying of solutions affects the convergence to and diversity in Pareto front.

Table 1. Comparison of Performance of ParRem1, ParRem2 and ParRem3 on Function UF1-UF9 with Population Size = 100 and Number of Generations = 3000

| Function | ParRem1 | | ParRem2 | | ParRem3 | | Sign test (IGD, SPREAD) |
|----------|----------------------|---------------------|----------------------|---------------------|----------------------|---------------------|-------------------------|
| | MeanIGD (Dying rate) | SPREAD (Dying rate) | MeanIGD (Dying rate) | SPREAD (Dying rate) | MeanIGD (Dying rate) | SPREAD (Dying rate) | |
| UF1 | 0.061657 (50%) | 0.58012 (50%) | 0.060779 (10%) | 0.55635 (50%) | 0.067689 (10%) | 0.55652 (50%) | (-, -) |
| UF2 | 0.019220 (50%) | 0.40815 (50%) | 0.023877 (50%) | 0.45033 (50%) | 0.022364 (10%) | 0.42707 (30%) | (+, +) |
| UF3 | 0.201176 (10%) | 0.52668 (10%) | 0.170225 (10%) | 0.47090 (50%) | 0.180125 (10%) | 0.50012 (30%) | (+, +) |
| UF4 | 0.050215 (10%) | 0.50465 (50%) | 0.0543763 (10%) | 0.53480 (50%) | 0.054034 (10%) | 0.55244 (30%) | (-, +) |
| UF5 | 0.153826 (10%) | 0.67871 (50%) | 0.208118 (10%) | 0.78428 (50%) | 0.201256 (10%) | 0.63878 (50%) | (+, +) |
| UF6 | 0.064901 (10%) | 0.70065 (50%) | 0.0704405 (10%) | 0.76884 (50%) | 0.059221 (30%) | 0.67963 (50%) | (+, +) |
| UF7 | 0.083477 (10%) | 0.67037 (50%) | 0.113827 (30%) | 0.79031 (50%) | 0.110274 (10%) | 0.70431 (50%) | (+, +) |
| UF8 | 0.755344 (50%) | 0.46400 (50%) | 0.738952 (50%) | 0.68068 (50%) | 0.738637 (30%) | 0.65029 (10%) | (-, +) |
| UF9 | 0.899907 (30%) | 0.63089 (50%) | 0.9045648 (10%) | 0.58913 (40%) | 0.959632 (30%) | 0.68203 (10%) | (+, +) |

From Table 1 it is observed that dying strategy ParRem1 with dying rate 50% has outperformed ParRem2 and ParRem3 in terms of SPREAD on four functions (UF2, UF4, UF7 and UF8) out of nine functions. ParRem2 has outperformed ParRem1 and ParRem3 on three functions (UF1 & UF3 with dying rate 50% and UF9 with dying rate 40%). ParRem3 has outperformed ParRem1 and ParRem2 on two functions (UF5 and UF6 with dying rate 50%). After analyzing the performance of the three strategies we have selected dying strategy ParRem1 and dying rate 50% for further experimentation.

Table 2. Statistical Sum of Problems for which each strategy obtains significantly better results

| Metric | ParRem1 | ParRem2 | ParRem3 |
|---------|---------|---------|---------|
| MeanIGD | 04 | 01 | 01 |
| SPREAD | 04 | 02 | 02 |
| Total | 08 | 03 | 03 |

From above discussion it is found that different strategies have given better performance on different functions. One strategy is not capable to solve all the optimization functions.

4 Ensemble of Dying Strategy Based MOGA (EDS-MOGA)

Ensemble learning is a machine learning paradigm where multiple learners are trained to solve the same problem. An ensemble combines a series of k learned models D_1, D_2, \dots, D_k , with the aim of creating an improved composite model D^* . A given set of solutions P is used to create k sets, P_1, P_2, \dots, P_k where D_i ($1 < i <= k-1$) is used to solve P_i . Ensemble is able to boost a weak learner to strong learner. Ensemble learning has proven to be very efficient and effective for adjusting algorithmic control parameters

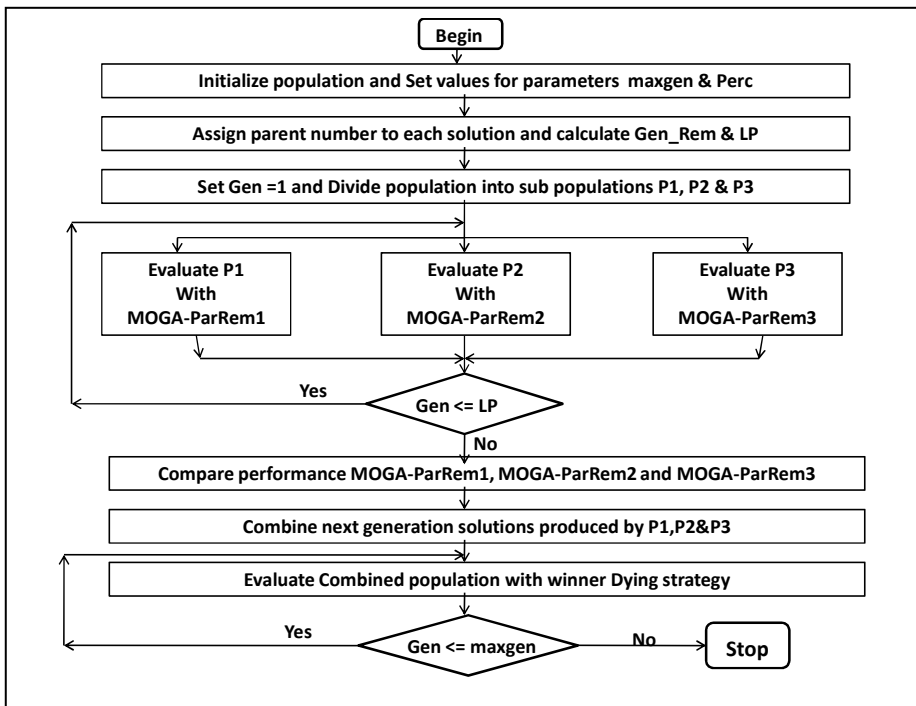


Fig. 1. Flowchart of EDS-MOGA

and operators in an online manner. Reference [9] proposes to use an ensemble of different Neighborhood Sizes in MOEA/D[18] and dynamically adjust their selection probabilities based on their previous performances. In the paper [16], differential evolution with an ensemble of restricted tournament selection (ERTS-DE) algorithm is introduced to perform multimodal optimization. It is impossible for a single constraint handling technique to outperform all other constraint handling techniques always on every problem irrespective of the exhaustiveness of parameter tuning. To overcome this selection problem, an ensemble of constraint handling methods (ECHM) to tackle constrained multi-objective optimization problems has been used. The ECHM is integrated with multi-objective differential evolution (MODE) algorithm [7]. In [17] authors propose an ensemble of mutation and crossover strategies and parameter values for DE (EPSDE) in which a pool of mutation strategies, along with a pool of values corresponding to each associated parameter competes to produce successful offspring population. Thus work done in [6]-[9] and [15]-[17] motivated authors to use ensemble methods to improve performance of MOGA with dying strategy.

Flowchart of proposed Ensemble of Dying Strategy Based Multi-Objective Genetic Algorithm (EDS-MOGA) is given in fig. 1. EDS-MOGA maintains a pool of three dying strategies. Population is divided into three sub populations. A Learning Period (LP) of few generation is defined and three subpopulations are evaluated with three dying strategy for LP generations. After LP generations’ performance of each subpopulation is compared and the strategy assigned to subpopulation having better objective function values will be selected as winner dying strategy. Three subpopulations are combined and the winner dying strategy will be used to evolve combined population for rest of the generations.

4.1 Experimentation and Results

Experiments are performed to demonstrate the behavior of the proposed approach in terms of convergence and diversity. Dying rate is chosen as 50% and other parameter setting used are same as given in section 3.

Table 3. Performance Comparison of EDS-MOGA with ParRem1, ParRem2 and ParRem3

| Scheme | ParRem1 | | ParRem2 | | ParRem3 | | EDS-MOGA | |
|----------|----------|---------|----------|---------|----------|---------|-----------------|----------------|
| Function | MeanIGD | SPREAD | MeanIGD | SPREAD | MeanIGD | SPREAD | MeanIGD | SPREAD |
| UF1 | 0.061657 | 0.58012 | 0.060779 | 0.55635 | 0.067689 | 0.55652 | 0.054910 | 0.55891 |
| UF2 | 0.019220 | 0.40815 | 0.023877 | 0.45033 | 0.022364 | 0.42707 | 0.013663 | 0.40647 |
| UF3 | 0.201176 | 0.52668 | 0.170225 | 0.47090 | 0.180125 | 0.50012 | 0.097539 | 0.49136 |
| UF4 | 0.050215 | 0.50465 | 0.054376 | 0.53480 | 0.054034 | 0.55244 | 0.016117 | 0.50529 |
| UF5 | 0.153826 | 0.67871 | 0.208118 | 0.78428 | 0.201256 | 0.63878 | 0.094909 | 0.53358 |
| UF6 | 0.064901 | 0.70065 | 0.070440 | 0.76884 | 0.059221 | 0.67963 | 0.060966 | 0.66912 |
| UF7 | 0.083477 | 0.67037 | 0.113827 | 0.79031 | 0.110274 | 0.70431 | 0.080293 | 0.66937 |
| UF8 | 0.755344 | 0.46400 | 0.738952 | 0.68068 | 0.738637 | 0.65029 | 0.271814 | 0.46076 |
| UF9 | 0.899907 | 0.63089 | 0.904564 | 0.58913 | 0.959632 | 0.68203 | 0.165976 | 0.55382 |

4.2 Results

MeanIGD and SPREAD values of UF1-UF9 produced by EDS-MOGA are reported in table 3. Table 3 also shows MeanIGD and SPREAD values of three dying strategies on UF1-UF9 function. EDS-MOGA has outperformed MOGA with single dying strategy i.e.ParRem1,ParRem2 and ParRem3 on functions UF1-UF5 & UF7-UF9 in terms of IGD metric and on functions UF2,UF5-UF9 functions in terms of SPREAD metric. EDS-MOGA combines' features of three dying strategy and contribution of each strategy has produced good convergence and diversity in the population. Dying strategies are dynamically selected by EDS-MOGA. The selection of appropriate strategy by EDS_MOGA has helped in improving performance of EDS-MOGA.

4.2.1 Comparison of EDS-MOGA with other Algorithms

Performance of EDS-MOGA is compared with NSGA-II [2] and SNOVMOGA (Summation of Normalized Objective Value based multi-objective Genetic algorithm) [19] algorithms. For fair comparison SBX operator in NSGA-II is replaced by MPX operator and mutation operator is also removed. NSGA-II-MPX is coded in MatLab 7.1. SNOVMOGA uses Summation of normalized objective value based sorting as discrimination technique while selecting parent solutions for formation of next generation population. In order to have diversity among solutions it uses reference point based improved selection method. Details of SNOVMOGA can be found in [19].

MeanIGD and SPREAD values of EDS-MOGA, NSGA-II-MPX and SNOVMOGA algorithms are reported in Table 4. EDS-MOGA has outperformed NSGA-II-MPX on functions UF1-UF6 &UF8-UF9. NSGA-II-MPX has given better MeanIGD and SNOVMOGA has given better SPREAD on UF7 function when compared to EDS-MOGA. Also SNOVMOGA has outperformed EDS-MOGA on function UF1 in terms of SPREAD metric.

Table 4. Performance Comparison of EDS-MOGA with NSGA-II-MPX and SNOVMOGA

| Scheme | EDS-MOGA | | NSGA-II-MPX | | SNOVMOGA | |
|----------|-----------------|----------------|-----------------|----------|----------|-----------------|
| Function | MeanIGD | SPREAD | MeanIGD | SPREAD | MeanIGD | SPREAD |
| UF1 | 0.054910 | 0.55891 | 0.068187 | 0.697345 | 0.101826 | 0.536142 |
| UF2 | 0.013663 | 0.40647 | 0.024311 | 0.540971 | 0.038288 | 0.428002 |
| UF3 | 0.097539 | 0.52136 | 0.135773 | 0.666309 | 0.316289 | 0.588970 |
| UF4 | 0.016117 | 0.50529 | 0.019737 | 0.682081 | 0.055698 | 0.631167 |
| UF5 | 0.094909 | 0.53358 | 0.211415 | 0.658579 | 0.593035 | 0.697120 |
| UF6 | 0.142566 | 0.66912 | 0.152526 | 0.694642 | 0.311779 | 0.684705 |
| UF7 | 0.080293 | 0.65037 | 0.031356 | 0.606167 | 0.066914 | 0.531024 |
| UF8 | 0.271814 | 0.46076 | 0.536727 | 0.616086 | 0.322564 | 0.793246 |
| UF9 | 0.165976 | 0.55382 | 0.375251 | 0.703306 | 0.108465 | 0.604505 |

Fig 2-9 shows plots of nondominated set obtained by EDS-MOGA and NSGA-II-MPX. Function UF5 is having discontinues Pareto-front. EDS-MOGA has successfully converged solutions on the Pareto-front of UF5 function whereas NSGA-II-MPX fails on UF5 as solutions are away from Pareto-front. EDS-MOGA has given poor spread on function UF7 as shown in fig. 6 where as NSGA-II-MPX has given good spread on UF7 as shown in fig.7. EDS-MOGA has shown better convergence and diversity when compared to NSGA-II-MPX.

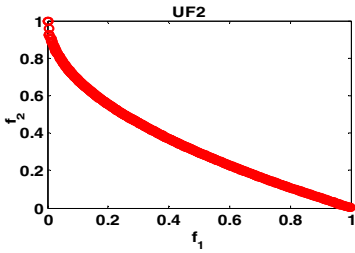


Fig. 2. Best approximate of UF2 with EDS-MOGA

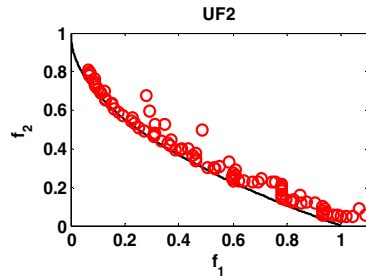


Fig. 3. Best approximate of UF2 with NSGA-II-MPX

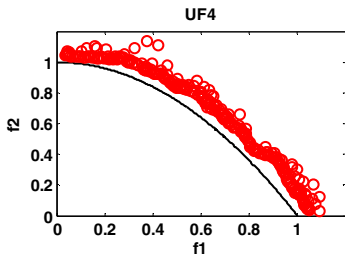


Fig. 4. Best approximate of UF4 with EDS-MOGA

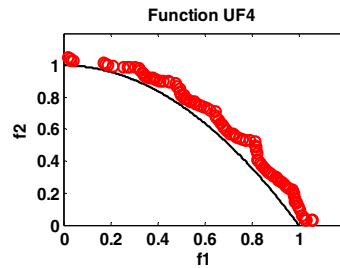


Fig. 5. Best approximate of UF4 with NSGA-II-MPX

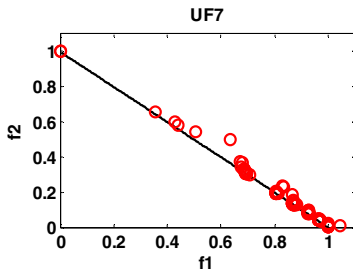


Fig. 6. Best approximate of UF7 with EDS-MOGA

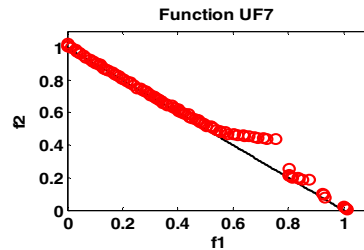


Fig. 7. Best approximate of UF7 with NSGA-II-MPX

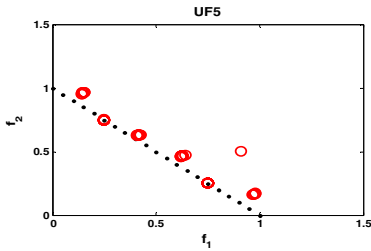


Fig. 8. Best approximate of UF5 with EDS-MOGA

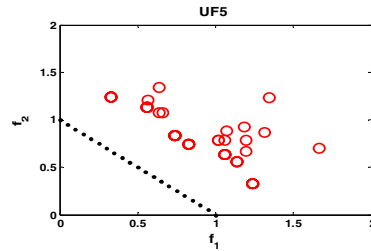


Fig. 9. Best approximate of UF5 with NSGA-II-MPX

5 Conclusion

This work has given insight into dying of solutions and its impact on performance of algorithm. Dying is implicit part of selection process i.e. solutions which are not selected die out. We have proposed and implemented strategies for explicit dying of solution in specific generations. Three strategies for dying of solution from next generation population have been proposed and implemented along with new MOGA framework. Experimental results indicate that the proposed MOGA with three strategies of dying of solutions is able to guide the search process towards the optimum for the seven bi-objective and the two 3-objective test functions. On comparing performance of three strategies of dying of solutions we found that no single strategy is capable enough to optimize all functions (UF1-UF9).

To improve performance of MOGA with dying strategy an ensemble of dying strategy is proposed and implemented. EDS-MOGA use all the three strategies of dying to evaluate solutions for few generations and then selects one dying strategy for remaining generations. Selection of dying strategy is based on performance of strategy in previous generations. EDS-MOGA has outperformed MOGA with single dying strategy. Also EDS-MOGA has shown better performance on many functions when compared to NSGA-II-MPX and SNOVMOGA. Thus ensemble of dying strategy has improved performance of MOGA.

Future work will be on fine tuning the selection of dying strategy and more strategy can be designed to have better convergence and diversity.

References

- [1] Deb, K.: Multi-objective Optimization using Evolutionary Algorithms. John Wiley & Sons, West Sussex (2001)
- [2] 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)

- [3] 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)
- [4] 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)
- [5] Al-Qunaieer, F.S., Tizhoosh, H.R., Rahnamayan, S.: Opposition Based Computing – A Survey. *IEEE Transactions on Evolutionary Computation* (2010), 978-1-4244-8126-2/10/ ©2010
- [6] Yu, E.L., Suganthan, P.N.: Ensemble of niching algorithms. *Inform. Sci.* 180(15), 2815–2833 (2000)
- [7] Mallipeddi, R., Suganthan, P.N.: Ensemble of constraint handling techniques. *IEEE Trans. Evol. Comput.* 14(4), 561–579 (2010)
- [8] Zhao, S.Z., Suganthan, P.N.: Multi-objective evolutionary algorithm with ensemble of external archives. *Int. J. Innovative Comput., Inform. Contr.* 6(1), 1713–1726 (2010)
- [9] Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Decomposition Based Multiobjective Evolutionary Algorithm with an Ensemble of Neighborhood Sizes. *IEEE Trans. on Evolutionary Computation* 16(3), 442–446 (2012)
- [10] Wolpert, D.H., Macready, W.G.: No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation* 1(1) (April 1997)
- [11] <http://en.wikipedia.org/wiki/Death>
- [12] <http://en.wikipedia.org/wiki/Extinction>
- [13] Tizhoosh, H.R.: Opposition-based learning: A new scheme for machine intelligence. In: Proceedings - International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA and International Conference on Intelligent Agents, Web Technologies and Internet, vol. 1, pp. 695–701 (2005)
- [14] 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)
- [15] Qu, B.-Y., Suganthan, P.N.: Novel Multimodal Problems and Differential Evolution with Ensemble of Restricted Tournament Selection. In: The Proc. IEEE Conference on Congress on Evolutionary Computation CEC 2010 (2010)
- [16] Qu, B.Y., Suganthan, P.N.: Constrained Multi-Objective Optimization Algorithm with Ensemble of Constraint Handling Methods. School of Electrical and Electronic Engineering Nanyang Technological University, Singapore Available at http://www.ntu.edu.sg/home/epnsugan/index_files/EEAs-EOAs.htm
- [17] Mallipeddi, R., Suganthan, P.N.: Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover Strategies. Nanyang Technological university, Singapore, Available at http://www.ntu.edu.sg/home/epnsugan/index_files/EEAs-EOAs.htm
- [18] Zhang, Q., Liu, W., Li, H.: The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: Proc. CEC, pp. 203–208 (2009)
- [19] Patel, R., Raghuwanshi, M.M., Malik, L.G.: An Improved Ranking Scheme For Selection Of Parents In Multi-Objective Genetic Algorithm. In: IEEE International Conference on Computer Security and Network Technology (CSNT) 2011, SMVDU Katra, Jammu (J&K), June 3-5, pp. 734–739 (2011)

Effect of Photovoltaic and Wind Power Variations in Distribution System Reconfiguration for Loss Reduction Using Ant Colony Algorithm

H.A. Abdelsalam¹, Almoataz Y. Abdelaziz²,
R.A. Osama², and Bijaya Ketan Panigrahi³

¹ Department of Electrical Engineering, Faculty of Engineering,
Kafrelsheikh University, Kafr Elsheikh, Egypt

² Department of Electrical Power & Machines, Faculty of Engineering,
Ain Shams University, Cairo, Egypt

³ Department of Electrical Engineering, Indian Institute of Technology, Delhi, India

Abstract. Intermittent characteristic of renewable power resources like photovoltaic (PV) power and wind power makes it very important to include power production at various times when evaluating the distribution system performance. This paper presents the effect of the intermittent renewable energy resources in the distribution system reconfiguration for loss reduction. The loss minimization problem is solved using the Ant Colony Optimization (ACO) algorithm implemented in the Hyper Cube (HC) framework. The 32-bus distribution network is studied for optimizing the configuration with and without the intermittent generations. The results of reconfiguration using the ACO algorithm show the improvement in the buses voltage profile with installing of PV and wind power sources with different values of solar irradiance and wind speed.

1 Introduction

Unpredicted variations in the electrical power generation from renewable energy resources are increasing day after day [1]. Wind and solar energy are both renewable energy and are also characterized by their intermittency [2, 3]. Intermittency means that they have both a non-controllable variability and are partially unpredictable. Non-controllable variability implies that an individual resource may be unavailable when needed. In fact wind turbines are really dependent of wind speed. Wind generation varies thus over time according to the wind speed. It is also very difficult to predict with accuracy wind's output even if the forecasting methods are improving. Photovoltaic (PV) panels have the same dependency towards the sunlight.

Distribution system Reconfiguration is the process of changing the topology of the distribution systems by changing the open/closed status of switches to transfer loads among the feeders. Two types of switches are used in primary distribution systems. There are normally closed sectionalizing switches and normally opened tie switches. Many advantages are obtained from feeder reconfiguration. Among them are real power loss reduction, balancing system load, bus voltage profile improvement, increasing system security and reliability, and power quality improvement. The concept of network reconfiguration for loss reduction was first introduced in reference [4].

Many reconfiguration techniques have been proposed which can be grouped into three main categories. One based on blend of heuristics and optimization techniques [5] which have very time consuming for large distribution systems therefore not practical for real time implementation. The second, based upon purely heuristics [6] where the optimality is not guaranteed and the algorithms are more likely to fall into local optimum. The third, techniques based on Artificial Intelligence and modern heuristics such as: Genetic Algorithms [7], Particle Swarm Optimization [8], Simulated Annealing [9], Tabu Search [10], etc. These AI based algorithms overcome the shortcoming of the conventional methods in saving the computation time, ensuring accuracy and optimality and thus suitable for real time applications [11]. There has been growing attention in algorithms inspired by the observation of natural phenomena to help solving complex combinatorial problems. In this paper, the Ant Colony Optimization (ACO) algorithm, proposed in [12], is employed to solve the reconfiguration problem. ACO algorithm is inspired by the foraging behavior of real ant colonies.

Authors of [13, 14, 15, 16] considered the output powers extracted from distributed generators (DGs), including the intermittent resources, as fixed values. As analyzed in [16], reconfiguration optimization problem of distribution network with DGs is solved using ACO and harmony search (HS) algorithms. ACO algorithm requires less practice to reach the optimum solution than HS Algorithm. This guarantees existing of the ACO solution. The ACO is considered in this paper to investigate the impact of the intermittent resources on the network reconfiguration at various times.

This paper is constructed as follows; section 2 presents the PV and wind turbine models as an intermittent energy sources. Section 3 explains the optimization problem of distribution system minimum loss reconfiguration. Section 4 illustrates the ACO algorithm for solving the network reconfiguration problem. Section 5 shows the study results of the 32-bus test systems. Finally the conclusion is given in Section 6.

2 Calculation of Photovoltaic and Wind Power

The Generated output power from the PV panel given by Eq. (1) depends on the area of PV panel A_s , solar irradiance coefficient μ and the efficiency of the PV panel Ω [17, 18]. Values of A_s and Ω can be obtained from manufacturer's data sheet.

$$P_{PV} = A_s \Omega \mu \quad (1)$$

The extracted power output from the wind turbine is given in Eq. (2) where ψ is the Albert Betz constant, ξ is the air density, A_w is area swept by turbine rotor, and v is the wind speed [17, 18]. Values of ψ , A_w and ξ can be obtained also from manufacturer's data sheet.

$$P_{wind} = \frac{1}{2} \psi A_w \xi v^3 \quad (2)$$

The intermittent parameters' values μ and v are taken from Table 1 [18]. The table shows the data of solar irradiance and wind speed data over 24 hours. Calculation of the power generated can be achieved at any irradiance level and wind speed by using the Eq. (1) and Eq. (2), respectively. In this work, the intermittent resources are considered as a constant power factor generation. The reactive power generation is calculated using the real power and the power factor for each PV panel and wind turbine.

Table 1. Intermittent parameters at different times during a day

| Time (hour) | Solar irradiance, μ (W/m ²) | Wind speed, v (m/s) |
|-------------|---|-----------------------|
| 1 AM | 0 | 3.03 |
| 6 AM | 32.18 | 6.39 |
| 11 AM | 872.77 | 4.32 |
| 6 PM | 139.98 | 6.07 |

3 Loss Reduction Reconfiguration Problem

Reconfiguration problem is formulated as follows:

$$\text{Minimize } f = \sum_{j=1}^{N_R} R_j |I_j|^2 \quad (3)$$

Subject to:

1. The bus voltage magnitude

$$V_{\min} \leq |V_i| \leq V_{\max}; \forall i \in N_b \quad (4)$$

2. The current limit of branches

$$|I_j| \leq I_{j \max}; \forall j \in N_R \quad (5)$$

3. Radial Topology

where f is the fitness function to be minimized corresponds to the total power loss in the system, R_j is the resistance of the branch j and I_j is the magnitude of the current flowing through the branch j , V_i is the voltage on bus i , V_{\min} and V_{\max} are minimum and maximum bus voltage limits respectively, I_j and $I_{j \max}$ are current magnitude and maximum current limit of branch j 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 [5]. This method has excellent convergence characteristics and is very robust and proved to be efficient for solving radial distribution networks.

To check the radiality constraints for a given configuration, a method based on the bus incidence matrix \hat{A} is used [19, 20] in which a graph may be described in terms of a connection or incidence matrix. Of particular interest is the branch to node incidence matrix \hat{A} , which has one row for each branch and one column for each node with a coefficient a_{ji} in row i and column j . The value of $a_{ji} = 0$ if branch j is not connected to node i , $a_{ji} = 1$ if branch j is directed away from node i and $a_{ji} = -1$ if branch j is directed towards node i . For a network calculation, a reference node must be chosen. The column corresponding to the reference node is omitted from \hat{A} and the resultant matrix is denoted by A . If the number of branches is equal to the number of nodes then, a square branch-to-node matrix is obtained. The determinant of A is then calculated. If $\det(A)$ is equal to 1 or -1 , then the system is radial. Else if the $\det(A)$ is equal

to zero, this means that either the system is not radial or group of loads are disconnected from service.

4 Ant Colony Algorithm

It is well known that the real ants are capable of finding the shortest path between their nest and food sources by the indirect communication between them via pheromone trails and this behavior forms the fundamental paradigm of the ant colony optimization algorithm [21].

ACO Paradigm

In the *ACO* method, a set of artificial ants cooperate in finding optimal solutions to difficult discrete optimization problems. These problems are represented as a set of points called *states* and ants move through adjacent states. Exact definitions of state and adjacency are problem specific. The *ACO* adopts three main rules:

1. The State Transition Rule

At first, each ant is placed on a starting state. Each will build a full path from the beginning to the end state through the repetitive application of the state transition rule, which is also called (“random proportional rule”).

$$P_k(i,j) = \frac{[\tau(i,j)]^\alpha [\eta(i,j)]^\beta}{\sum_{m \in J_k(i)} [\tau(i,m)]^\alpha [\eta(i,m)]^\beta}, \forall j \in J_k(i) \quad (6)$$

where, $P_k(i,j)$ is the probability with which ant k in node i chooses to move to node j , $\tau(i,j)$ is the pheromone which deposited on the edge between nodes i and j , $\eta(i,j)$ is the visibility of the edge connecting nodes i and j which is problem specific (e.g. inverse of the edge distance), $J_k(i)$ is the set of nodes that remain to be visited by ant k positioned on node i . α and β are parameters that determine the relative importance of pheromone versus the path's visibility. The state transition rule favors transitions toward nodes connected by shorter edges with greater amount of pheromone.

2. Local Pheromone Updating Rule

While constructing the solution, each ant modifies the pheromone on the visited path. It is an optional step intended to shuffle the search process. It increases the exploration of other solutions by making the visited lines less attractive.

$$\tau(i,j) = (1 - \rho)\tau(i,j) + \rho\tau_0 \quad (7)$$

where $\tau(i,j)$ is the amount of pheromone deposited on the path connecting nodes i and j , τ_0 is the initial pheromone value and ρ is a heuristically defined parameter.

3. Global Pheromone Updating Rule

When all tours are completed, the global updating rule is applied to edges belonging to the best ant tour providing a greater amount of pheromone to shortest tour.

$$\tau(i, j) = (1 - \sigma)\tau(i, j) + \sigma\delta^{-1} \quad (8)$$

where δ is a parameter belonging to the globally best tour and σ is the pheromone evaporation factor element in the interval [0-1]. This rule is intended to make the search more directed enhancing the capability of finding the optimal solution.

Formulation of ACO in the HC Framework

The HC framework is a recently developed framework for the standard ACO [22, 23]. It is based on changing the pheromone update rules used in ACO algorithms so that the range of pheromone variation is limited to the interval [0-1], thus providing automatic scaling of the auxiliary fitness function used in the search process and resulting in a more robust and easier to implement version of the ACO procedure. The distribution system is represented as an undirected graph $G(B, L)$ composed of set B of nodes and a set L of arcs indicating the buses and their connecting branches (switches) respectively. Artificial ants move through adjacent buses, selecting switches that remain closed to minimize the system power losses. The solution iterates over three steps:

1. Initialization:

The Solution starts with encoding (i) system parameters such as; set of supply substations S ; set of buses NB ; set of branches NR where each branch has 2 possible states either "0" for an opened tie switch or "1" for a closed sectionalizing switch); load data $Pload, Qload$; branch data Rb, Xb ; base configuration of the system $C(0)$ defined by the system's tie switches, initial power losses of the system $f(C(0))$ by solving the power flow for $C(0)$ and evaluating the fitness function f and (ii) algorithm parameters such as; number of artificial ants in each iteration N ; initial pheromone quantity τ_0 assigned to each switch; evaporation factor of pheromone trails ρ ; the parameters α and β that determine the relative importance of the line's pheromone versus its visibility; a counter h for the number of iterations, a counter x that is updated at the end of the iteration with no improvement in the objective function; maximum number of iterations $Hmax$, and maximum number of iterations $Xmax$ with no improvement in the objective function respectively. The base configuration is then set as an initial reference configuration and as the best configuration found so far such that $C_{best} = C_{best}(0) = C(0)$.

2. Ants' Reconfiguration and Pheromone Updating

In each iteration h , a reference configuration is set as the best configuration of the previous iteration such that $C_{best}(h-1) = C_{ref}(h)$. N Ants are initially located on N randomly chosen open switches and are sent in parallel in such a way that each ant n in the h th iteration introduces a new radial configuration $C_n(h)$ by applying the state transition rule. Once all ants finish their tour, the configuration corresponding to each ant is evaluated by computing the objective function $f(C_n(h))$. The best configuration of the h th iteration $C_{best}(h)$ is identified which is the configuration corresponds to the minimum evaluated objective function of all ants (minimum power loss). The best

configuration of the h th iteration $C_{best}(h)$ is compared to the best configuration so far C_{best} such that if $f(C_{best}(h)) < f(C_{best})$, the overall best configuration is updated such that $C_{best} = C_{best}(h)$. Finally, the pheromone updating rules are applied such that for all switches that belong to the best configuration, the pheromone values are updated using Eq. (9). Otherwise, the pheromone is updated using Eq. (10).

$$\tau^{(h)} = (1 - \rho)\tau^{(h-1)} + \rho\sigma \quad (9)$$

$$\tau^{(h)} = (1 - \rho)\tau^{(h-1)} \quad (10)$$

where, $\tau^{(h)}$ is the new pheromone value after the h^{th} iteration, $\tau^{(h-1)}$ is the old value of pheromone after the $(h^{\text{th}} - 1)$ iteration, ρ is arbitrarily chosen from the interval $[0-1]$ and σ is a heuristically defined parameter which was chosen to be equal $(f(C_{best}) / f(C_{best}^{(h)}))$ since $f(C_{best}^{(h)})$ cannot be lower than $f(C_{best})$ the pheromone assigned to any switch cannot fall outside the range $[0-1]$ so that the pheromone update mechanism is fully consistent with the requirements of the HC framework.

3. Termination Criteria

The solution process continues until maximum number of iterations is reached $h=H_{max}$, or until no improvement of the objective function has been detected after specified number of iterations $x=X_{max}$.

5 Numerical Results and Simulations

In this section, the 32-bus system is tested. The system is a 12.66 kV radial distribution system whose data is given in [6]. The system has one supply point, 32 buses, 3 laterals and 5 tie switches. The total substation loads of the initial configuration are 3715 kW and 2300 kVAR. The base configuration of the system is shown in Fig. 1 and can be defined by the system's tie switches 13, 37, 22, 26 and 29. It has real power loss of 203 kW. The HC-ACO parameters used are $N=10$, $\alpha=0.1$, $\beta=0.9$, $\rho=0.04$, $\tau_0=1$, $H_{max}=100$ and $W_{max}=10$. Four intermittent generators are installed as shown in Table 2 based on the study in [24]. In this work, PV panels are installed at buses 2 and 5 whereas wind turbines inserted at buses 23 and 28. The following ten cases are: considered as follows:

1. Base configuration of the system without intermittent resources.
2. Reconfiguration of case (1) using the ACO algorithm.
3. Including PV and wind energy in the base configuration of the system at 6:00 AM.
4. Reconfiguration of case (3) using the ACO algorithm.
5. Including PV and wind energy in the base configuration of the system at 11:00 AM.
6. Reconfiguration of case (5) using the ACO algorithm.
7. Including PV and wind energy in the base configuration of the system at 6:00 PM.
8. Reconfiguration of case (7) using the ACO algorithm.
9. Including PV and wind energy in the base configuration of the system at 1:00 AM.
10. Reconfiguration of case (9) using the ACO algorithm.

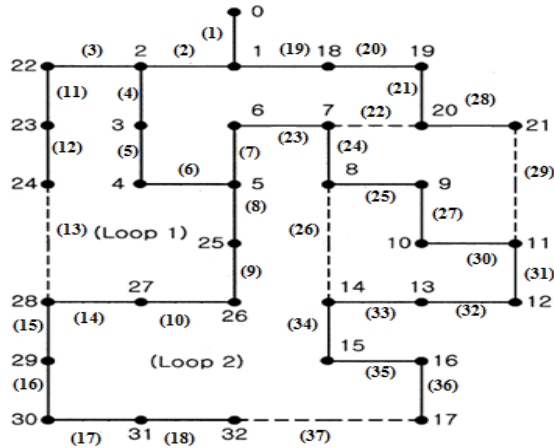


Fig. 1. Base configuration of the 32-bus system. Tie switches are represented by dotted lines.

Table 2. Installation buses of PV array and wind farm

| Bus # | 2 (PV array) | 5 (PV array) | 23 (wind farm) | 28 (wind farm) |
|-------|--------------|--------------|----------------|----------------|
| P.F. | 0.9 | 0.9 | 0.9 | 0.9 |

Intermittent power (in Table 3) is calculated using Eq. (1) and Eq. (2) based on the manufacturing data sheets [25, 26, 27] and the coefficients in Table 1 for the cases 3, 4, 5, 6, 7, 8, 9 and 10. From the manufacturing data sheets [25, 26], PV module is taken as 36 cells with 3 modules per panel. This work considers PV array of 372 panels. The PV module has an efficiency of 15.4 % and area of 1.656 m². The wind turbine [27] has Albert Betz constant of 0.4, air density of 1.23 and radius of turbine rotor of 10 m. The wind farm is considered in this work contains 30 wind turbines.

Table 3. Power generated from PV arrays and wind farms

| | Case 3,4 | Case 5,6 | Case 7,8 | Case 9,10 |
|--------------------------------------|----------|----------|----------|-----------|
| Solar Irradiance (W/m ²) | 32.18 | 872.77 | 139.98 | 0 |
| Wind speed (m/s) | 6.39 | 4.32 | 6.07 | 3.03 |
| P _{PV} at node#2 (kW) | 9.14 | 247.86 | 39.75 | 0 |
| Q _{PV} at node#2 (kVAR) | 4.43 | 120.05 | 19.25 | 0 |
| P _{PV} at node#5 (kW) | 9.14 | 247.86 | 39.75 | 0 |
| Q _{PV} at node#5 (kVAR) | 4.43 | 120.05 | 19.25 | 0 |
| P _{Wind} at node#23 (kW) | 605.33 | 187.04 | 518.86 | 64.54 |
| Q _{Wind} at node#23 (kVAR) | 293.17 | 90.58 | 251.29 | 31.26 |
| P _{Wind} at node#28 (kW) | 605.33 | 187.04 | 518.86 | 64.54 |
| Q _{Wind} at node#28 (kVAR) | 293.17 | 90.58 | 251.29 | 31.26 |

Tables 4 and 5 show the simulation results of the ten cases. As shown, tie switches of the system before the reconfiguration (Table 4) are 13, 37, 22, 26 and 29 with minimum power loss of 103 kW corresponding to case 3. The cases in Table 5 are corresponding to the reconfiguration of the cases in Table 4. The final tie switches of cases in Table 5 are different than the corresponding cases in Table 4. Also the configurations of cases (2, 4, 6, 8 and 10) are different. These results imply that due to the intermittent nature of the renewable energy resources installed in the distribution system, the configuration of the network changed (as illustrated in Tables 4 and 5) which makes the real time reconfiguration of the distribution network necessary. Real time reconfiguration can be achieved by sending suitable commands to the tie switches as a part of the smart distribution grid.

Table 4. Effect of PV and wind power before reconfiguration

| | Case 1 | Case 3 | Case 5 | Case 7 | Case 9 |
|------------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Base Tie Switches | 13 37 22 26 29 | 13 37 22 26 29 | 13 37 22 26 29 | 13 37 22 26 29 | 13 37 22 26 29 |
| P_{loss} (kW) | 203 | 103 | 137.2 | 110.7 | 189.9 |
| LBV | 0.9129 at bus #17 | 0.9287 at bus #17 | 0.9240 at bus #17 | 0.9273 at bus #17 | 0.9146 at bus #17 |

Table 5. Effect of PV and wind power after reconfiguration

| | Case 2 | Case 4 | Case 6 | Case 8 | Case 10 |
|------------------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Final Tie Switches | 13 18 23 33 27 | 14 36 22 33 25 | 13 18 23 25 33 | 10 37 22 25 33 | 14 18 23 33 25 |
| P_{loss} (kW) | 139.5 | 65.8 | 96 | 71.9 | 129.7 |
| LBV | 0.9378 at bus #31 | 0.9604 at bus #17 | 0.9478 at bus #32 | 0.9597 at bus #32 | 0.9444 at bus #31 |

The voltage profile of different cases of the system is shown in Fig. 2 to Fig. 6. Low bus voltage (LBV) of the initial configuration of the system (cases of Table 4) is varied according to the intermittency of the PV panels and wind turbines but still occur at the same bus # 17. In cases 4 and 8, most of the node voltages have been improved after reconfiguration as shown in Fig. 3 and Fig. 5, respectively. In case 4, the LBV value is that of bus 17 equals to 0.9604 p.u. achieving 3.41% improvement than case 3. In case 8, the LBV value is that of bus 32 equals to 0.9597 p.u. achieving 3.49% improvement than case 7. Fig. 3 to Fig. 6 illustrate that the voltage profile improvement of the reconfigured network depends on the change of the PV and wind power.

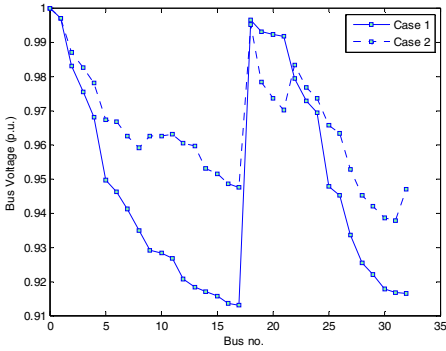


Fig. 2. Voltage profile for cases 1 and

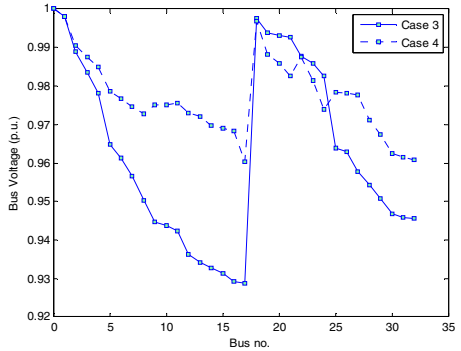


Fig. 3. Voltage profile for cases 3 and 4

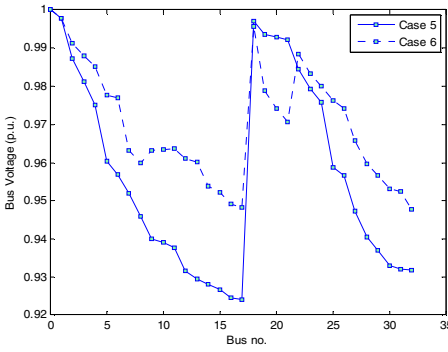


Fig. 4. Voltage profile for cases 5 and 6

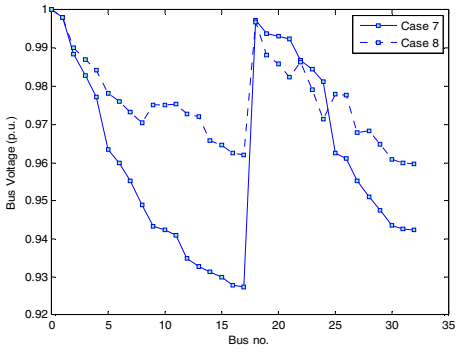


Fig. 5. Voltage profile for cases 7 and 8

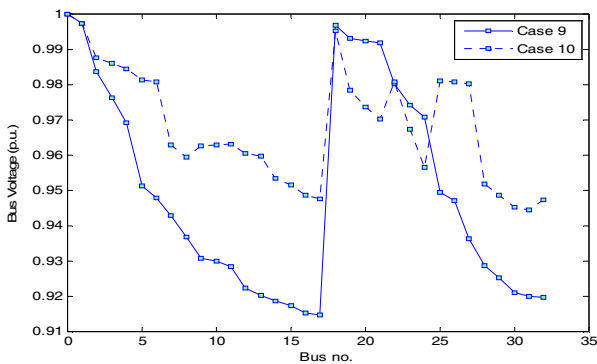


Fig. 6. Voltage profile for cases 9 and 10

6 Conclusion

This paper studied the reconfiguration of a distribution network with intermittent energy plants for loss minimization using ACO method. ACO optimization algorithm was used to find the most appropriate topology of the distribution system with and without intermittent resources that minimize the total system power loss. The objective function is subjected to many constraints such as bus voltage limits, branch current limits and radial configuration format. The simulation results showed how to involve the intermittent renewable energy resources in the distribution system considering the reconfiguration process. The solar and wind power were calculated based on manufacturers' information. Then different cases were studied to show the effect of the generated power variation on the resulting tie switches, minimum losses and low bus voltage. Insertion of PV panels and wind turbine to the study distribution system reduced the total power loss and improved bus voltage profile. Losses were further reduced by reconfiguration. Unpredicted nature of the renewable energy changed the system topology which needs a real time reconfiguration in order to keep minimum loss of distribution system over different weather conditions.

References

1. Liang, J., Molina, D., Venayagamoorthy, G.K., Harley, R.G.: Two-Level Dynamic Stochastic Optimal Power Flow Control for Power Systems with Intermittent Renewable Generation. *IEEE Transactions on Power Systems* (9) (January 2013)
2. Ochoa, L.F., Dent, C.J., Harrison, G.P.: Maximisation of intermittent distributed generation in active networks. In: *Proc. 2008 CIRED Seminar: SmartGrids for Distribution* (2008)
3. Ochoa, L.F., Harrison, G.P.: Minimizing Energy Losses: Optimal Accommodation and Smart Operation of Renewable Distributed Generation. *IEEE Transactions on Power Systems* 26(1), 198–205 (2011)
4. Merlin, A., Back, H.: Search for a minimal-loss operating spanning tree configuration in an urban power distribution system. In: *Proc. 5th Power System Computation Conf. (PSCC)*, Cambridge, UK, pp. 1–18 (1975)
5. Shirmohammadi, D., Hong, H.W.: A compensation-based power flow method for weakly meshed distribution and transmission networks. *IEEE Trans. Power Systems* 3, 753–762 (1988)
6. Baran, M.E., Wu, F.F.: Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Trans. Power Delivery* 4, 1401–1407 (1989)
7. 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 Systems* 23, 186–195 (2008)
8. Abdelaziz, A.Y., Mohamed, F.M., Mekhamer, S.F., Badr, M.A.L.: Distribution systems reconfiguration using a modified particle swarm optimization algorithm. *Electric Power Systems Research* 79, 1521–1530 (2009)
9. Chiang, H.D., Jumeau, R.J.: Optimal network reconfigurations in distribution systems: Part 2. *IEEE Transactions Power Delivery* 5, 1568–1575 (1990)

10. Abdelaziz, A.Y., Mohamed, F.M., Mekhamer, S.F., Badr, M.A.L.: Distribution system re-configuration using a modified Tabu Search algorithm. *Electric Power Systems Research* 80, 943–953 (2010)
11. Bouchard, D.E.: Towards Loss Minimization in power distribution systems using AI: the WatDist Algorithm. Ph.D. dissertation, University of Waterloo, Ontario, Canada (1996)
12. Dorigo, M.: Optimization, learning and natural algorithms. Ph.D. dissertation, Dep. Elec. Eng., Univ. of Milano, Italy (1992)
13. Sarfi, R.J., Salama, M.M.A., Chikhani, A.Y.: A survey of the state of the art in distribution system reconfiguration for system loss reduction. *Electric Power Systems Research* 31, 61–70 (1994)
14. Yasin, Z.M., Rahman, T.K.A.: Service Restoration in Distribution Network with Distributed Generation. In: Proc. 2006 IEEE 4th Student Conf. of Research and Development, Shah Alam, Selangor, Malaysia, pp. 200–205 (2006)
15. Rugthaicharoencheep, N., Sirisumrannuku, S.: Feeder Reconfiguration for Loss Reduction in Distribution System with Distributed Generators by Tabu Search. *GMSARN International Journal* 3, 47–54 (2009)
16. Abdelaziz, A.Y., Osama, R.A., Elkhodary, S.M., El-Saadany, E.F.: Reconfiguration of Distribution Systems with Distributed Generators using Ant Colony Optimization and Harmony Search Algorithms. In: Proceedings of the 2012 IEEE PES General Meeting, San Diego, California, USA, July 22–26 (2012)
17. Saber, A.Y., Venayagamoorthy, G.K.: Plug-in Vehicle and Renewable Energy Sources for Cost and Emission Reductions. *IEEE Transaction on Industrial Electronics* 58(4) (April 2011)
18. Yammani, C., Maheswarapu, S., Matam, S.: Optimal Placement of Multi DGs in Distribution System with Considering the DG Bus Available Limits, <http://article.sapub.org/10.5923.j.ep.20120201.03.html>
19. Wright, P.: On minimum spanning trees and determinants. *Mathematics Magazine* 73, 21–28 (2000)
20. Grainger, J., Stevenson Jr., W.: *Power System Analysis*. McGraw-Hill, New York (1994)
21. Dorigo, M., Stutzle, T.: *Ant Colony Optimization*. A Bradford book, pp. 1–44. Massachusetts Institute of Technology (2004)
22. Carpaneto, E., Chicco, G.: Distribution system minimum loss reconfiguration in the HC-AC Optimization framework. *Electric Power Systems Research* 78, 2037–2045 (2008)
23. Blum, C., Dorigo, M.: The hyper cube framework for ant colony optimization. *IEEE Trans. System Man and Cybernetics* 34, 1161–1172 (2004)
24. Wu, Y.-K., Lee, C.-Y., Liu, L.-C., Tsai, S.-H.: Study of Reconfiguration for the Distribution System with Distributed Generators. *IEEE Trans. Power Delivery* 25(3) (July 2010)
25. http://www.isofoton.com/sites/default/files/255_en__2.pdf
26. <http://www.solardirect.com/pv/pvlist/pvlist.htm>
27. http://www.raeng.org.uk/education/.../23_wind_turbine.pdf

Inter-species Cuckoo Search via Different Levy Flights

Swagatam Das¹, Preetam Dasgupta², and Bijaya Ketan Panigrahi³

¹ Indian Statistical Institute, Kolkata, India

² Jadavpur University, Kolkata, India

³ Indian Institute of Technology

Abstract. In this paper we improve the meta heuristic algorithm known as Cuckoo Search (CS) to solve optimization problems. The proposed Inter-species Cuckoo Search (ISCS) algorithm is based on the brood parasitic behavior of different inter-related cuckoo species in different areas in combination with Levy flight behavior(which changes with the terrain) of birds. The proposed algorithm is then tested against various test functions and its performance is compared with genetic algorithms, particle swarm optimization and previous versions of Cuckoo Search algorithm.

Keywords: clustering, inter-species cuckoo search, Levy flight, metaheuristics, optimization.

1 Introduction

Modern meta-heuristic algorithms are more and more inspired by natural entities and as an example we see various different swarm intelligence methods emerging. For example cuckoo search (CS) algorithm [1] is inspired by brooding behavior of cuckoo, while the bat algorithm [3] comes forth from echolocation behavior of bats, firefly algorithm [4] was inspired by flashing pattern of tropical fireflies and particle swarm optimization (PSO) [5] was inspired by fish and bird swarm intelligence found in nature. The main reason behind the emergence and success of different meta-heuristic methods is the proper imitation of nature and mostly biological systems i.e. the presence of acute intensification and diversification [6]. Some very modern and advanced works on this topic are done in [25]. Cuckoo search has some very modern applications in discrete domain as well i.e. in job scheduling problems [26, 27].

Cuckoo Search via Levy Flights [1] is a novel meta-heuristics algorithm which is based on the interesting breeding behavior such as brood parasitism of certain species of cuckoos. The basic ideas applied are the aggressive reproduction strategy of cuckoo and usage of Levy flights. Cuckoo Search algorithm is being widely used in engineering optimization problems [2] with exceptionally good results. However this method has space for improvement and here we improve Cuckoo Search by dividing the entire breeding ground in different terrains via clustering and using different types of Levy flights for different areas. This change not only reduces the optimization time but also increases the search accuracy. Simulation results show that the improved algorithm has better optimization ability.

2 Basic Cuckoo Search Algorithm

2.1 Cuckoo Breeding Behavior

Cuckoo are fascinating birds, not only because of the beautiful sounds they can make, but also because of their aggressive reproduction strategy. Some species such as the ani and Guira cuckoos lay their eggs in communal nests, though they may remove others' eggs to increase the hatching probability of their own eggs [12]. A number of species engage the obligate brood parasitism by laying their eggs in the nests of other host birds (often other species). There are three basic types of brood parasitism: intra-specific brood parasitism, cooperative breeding, and nest takeover. Some host birds can engage in direct conflict with the intruding cuckoos. If a host bird discovers the eggs are not their owns, they will either throw these alien eggs away or simply abandon the nest and build a new nest elsewhere. Some cuckoo species such as the New World brood-parasitic *Tapera* have evolved in such a way that female parasitic cuckoos are often very specialized in the mimicry in colour and pattern of the eggs of a few chosen host species [12]. This reduces the probability of their eggs being abandoned and thus increases their reproductivity.

2.2 Levy Flights

Various studies have shown that flight behavior of many animals and insects has demonstrated the typical characteristics of Levy flights [4, 15, 13, 14]. A recent study by Reynolds and Frye shows that fruit flies or *Drosophila melanogaster*, explore their landscape using a series of straight flight paths punctuated by a sudden 90° turn, leading to a Lévy-flight-style intermittent scale free search pattern. Studies on human behaviour such as the Juhoansi hunter-gatherer foraging patterns also show the typical feature of Levy flights. Even light can be related to Levy flights [1]. Subsequently, such behaviour has been applied to optimization and optimal search, and preliminary results show its promising capability [13, 15, 19, 20].

Cuckoo Search

The basic version of Cuckoo Search (CS) explained by Xin-She Yang and Suash Deb [1] has the following rules:

- 1) Each cuckoo lays one egg at a time, and dump its egg in randomly chosen nest;
- 2) The best nests with high quality of eggs will carry over to the next generations;
- 3) The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in [0, 1]$. The last assumption can be approximated by the fraction of the n nests that are replaced by new nests with random solutions.

For a maximization problem, the fitness of the solution can simply be proportional to the objective function. Other forms of fitness can be defined in a similar way to the fitness function in genetic algorithms.

Here each egg in a nest represents a solution, and each cuckoo egg represents a new solution, the aim is to use the new and potentially better solutions (cuckoos) to replace a not-so-good solution in the nests.

The basic steps of Cuckoo Search (CS) are shown as a part of the proposed algorithm in Fig. 1.

When generating new solutions $x^{(t+1)}$ for, say cuckoo i , a Levy flight is performed

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Levy(\lambda), \quad (1)$$

where $\alpha > 0$ is the step size which is related to the scales of the problem of interest. In most cases, it was taken as

$\alpha = O(1)$. The product \oplus means entry-wise multiplications. Levy flights essentially provide a random walk while their random steps are drawn from a Levy distribution

$$Levy \sim u = t^{-\lambda}, \quad (1 < \lambda \leq 3), \quad (2)$$

which has an infinite variance with an infinite mean. Here the consecutive jumps/steps of a cuckoo essentially form a random walk process which obeys a power-law step-length distribution with a heavy tail.

3 Inter-species Cuckoo Search

In case of multimodal situations the basic Cuckoo Search may not be able to find all the optima and as this search uses only Yang-Deb levy as Levy flight method it might lead to less robustness and unable to search the whole region effectively.

In our proposed algorithm known as the Inter-species Cuckoo Search (ISCS) we improve basic Cuckoo Search such that it can perform exceptionally sound even in multimodal situations and with more robustness.

Here we divide the whole search space in various regions or terrains and disperse the host nests equally in these regions. Then we assume different kinds of cuckoo are present in different regions and their random walk behavior are also different. Thus, different types of cuckoo species are present in the total search space. The generation of new solutions around the best solutions is dependent upon the levy flights and since different species of cuckoo exhibit different types of Levy flight the Inter-species Cuckoo Search can explore the search space more effectively and can perform much better local search than the basic version of cuckoo search in both unimodal and multimodal situations. A substantial fraction of the new solutions are generated by far field randomization. The cuckoo in different regions communicate with each other by replacing some of the worst solutions or nests of one region with the best solutions of other regions after a fixed interval of time i.e. after a certain no of function evaluations. The most important features of the proposed algorithm over basic Cuckoo Search are explained afterwards.

3.1 Clustering

The main idea behind Inter-species Cuckoo Search (ISCS) is the division of entire search space into various regions or terrains. This crucial step is performed by clustering the initial search space into various clusters and treating each cluster as a habitat only for a particular species of cuckoo. Here we have used the well known

“k-means” algorithm for clustering but any clustering method would suffice. Here the number of clusters or regions was kept to a fixed value but it can also be variable or adaptive if needed.

3.2 Different Levy Flights

A Lévy flight is a random walk in which the step lengths have a probability distribution that is heavy-tailed. When defined as a walk in a space of dimension greater than one, the steps made are in isotropic random directions. The variation of levy flights is done on the basis of distribution of step sizes. The variants are explained below,

- Yang-Deb levy:

This variant was used by Xin-She Yang and Suash Deb [1] and in this case the random distribution steps are drawn from a Levy distribution

$$Levy \sim u = t^{-\lambda}, \quad (1 < \lambda \leq 3), \quad (3)$$

which has an infinite variance with infinite mean. The consecutive jumps or steps form a random walk process which obeys a power-law step length distribution with a heavy tail.

- Cauchy Levy flight:

Here the step sizes of Levy flights are drawn from Cauchy distribution or Cauchy-Lorentz distribution. It is a continuous probability distribution and It has the distribution of a random variable that is the ratio of two independent standard normal random variables. This has the probability density function

$$f(x;0,1) = \frac{1}{\pi(1+x^2)}. \quad (4)$$

Thus the distribution of step size is done by the simplest form of Cauchy distribution known as standard Cauchy distribution.

- Rayleigh Levy flight:

In this type of Levy flight the distribution of step sizes is Normal distribution, which is a continuous probability distribution, defined by the formula

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5)$$

The parameter μ in this formula is the mean or expectation of the distribution (and also its median and mode). The parameter σ is its standard deviation; its variance is therefore σ^2 .

- Gaussian walk:

In some cuckoo species i.e. in some regions the distribution of step size in Levy flights follow Gaussian walk which is an inverse cumulative normal distribution.

3.3 Information Sharing

In order to explore the search space effectively and to reduce computation a method of data sharing among cuckoo of different region or species is used. After a certain number of generations the worst 10 nests of a species or region are replaced by 10 nests taken randomly from the best 10 nests of every other cuckoo species. This step

makes the algorithm more robust and a certain amount of far field randomization is also covered by this step.

Inter-species Cuckoo Search

begin

Objective function $f(x)$, $x = (x_1, \dots, x_d)^T$

Generate initial population of

n host nests x_i ($i = 1, 2, \dots, n$)

Divide the entire population in 4 clusters using k-means algorithm (each cluster contains $n/4$ nests).

while ($t < \text{MaxGeneration}$) or (stop criterion)

for each cluster do,

if (cluster number == 1),

 apply Yang-Deb levy flight;

end

if (cluster number == 2),

 apply Cauchy Levy flight;

end

if (cluster number == 3),

 apply Rayleigh flight;

end

if (cluster number == 4),

 apply Gaussian walk;

end

Get a cuckoo randomly by Levy flights

Evaluate its quality/fitness F_i

Choose a nest among n (say, j) randomly

If ($F_i > F_j$),

 Replace j by the new solution;

end

A fraction (P_a) of worse nests are abandoned and new ones are built;

Keep the best solutions (or nests with quality solutions);

Rank the solutions and find the current best for each cluster,

 After every 10 generations,

 For each cluster do,

 Replace 10 worst nests of the cluster by a random set of 10 best nests of every other cluster.

 end while

find global best solution

end

Fig. 1. Pseudo code of the Inter-species Cuckoo Search (ISCS)

Thus, in Inter-species Cuckoo Search (ISCS) the entire search space is divided in different terrains in which different species of cuckoo exist. In each region a basic version of cuckoo search is performed with different types of distributions are used to determine the step size of Levy flights. In each region some of the new solutions are generated by levy flights around the best solution obtained so far, this is the main local search performed within the cluster or region. However, a substantial fraction of new solutions are generated by far field randomization within the cluster radius i.e. within the species but whose locations are far enough from the current best solution in the region.

The presence of different species of cuckoo in different regions makes the algorithm suitable for any multimodal scenario and there exists a certain amount of data/information sharing among the different species. This data sharing is done by replacement of a certain number of worst nests of a region by a random set created from a certain number of best nests of all other regions. This information sharing creates the opportunity of a finer search around the local best solutions found in various clusters and provides a better chance to reach the global optimum.

4 Implementation and Numerical Experiments

4.1 Validation and Parameter Settings

To test the performance of the proposed algorithm the Inter-species Cuckoo Search was coded in Matlab and run on a PC with an Intel Core Duo 2.4GHz CPU, 4GB RAM, running Windows XP.

In Fig. 2, Fig. 3 and Fig. 4 we show how the cuckoos have explored the entire search space and within 10 generations they have clustered around the global best solution. Afterwards the exact global position was found as a result of the finer local search performed by different cuckoo species around the best solution.

Here we have used $n=40$ nests divided into 4 clusters or regions, i.e. 10 nests for each cuckoo species. In most of our simulations we have used $\alpha = 1$ and $p_a = 0.25$ and $n=40$ to 60.

From the figure, we can see that, as the optimum is approaching, most nests aggregate towards the global optimum. Also the nests are distributed throughout the search space but within 10 generations they cluster around the global best. The presence of different types of cuckoo species makes the algorithm useful in multimodal and multi-objective optimization problems.

The number of host nests i.e. the population size n was varied from 20 to 400 in steps of 20 but for most optimization problems we found $n=40$ or $n=60$ would produce decent results. The probability p_a was chosen randomly for each cluster and these values were between 0.1 and 0.5. No parameters were changed in the expressions of

different levy flights. Thus, no fine adjustment was needed for any specific problems and so the Inter-species Cuckoo Search (ISCS) proves to be a very much generalized heuristic algorithm with a broader range of possible applications.

4.2 Test Function

Any new optimization algorithm should also be validated and tested against these benchmark functions. In our simulations, we have used the following test functions.

De Jong’s first function is essentially a sphere function

$$f(x) = \sum_{i=1}^d x_i^2, \quad x_i \in [-5.12, 5.12], \tag{7}$$

Whose global minimum $f_* = 0$ occurs at $x_* = (0, 0, \dots, 0)$.

Here d is the dimension.

Easom’s test function is unimodal

$$f(x, y) = -\cos(x)\cos(y)\exp[-(x-\pi)^2 - (y-\pi)^2], \tag{8}$$

Where

$$(x, y) \in [-100, 100] \times [-100, 100]. \tag{9}$$

It has a global optimum of $f_* = -1$ at (π, π) in a very small region.

Shubert’s bivariate function

$$f(x, y) = -\sum_{i=1}^5 i \cos[(i+1)x + 1] \sum_{i=1}^5 \cos[(i+1)y + 1], \tag{10}$$

has 18 global minima in the region

$$(x, y) \in [-10, 10] \times [-10, 10].$$

The value of its global minima is $f_* = -186.7309$.

Griewangk’s test function has many local minima

$$f(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \tag{11}$$

but a single global minimum $f_* = 0$ at $x_* = (0, 0, \dots, 0)$ for all

$-600 \leq x_i \leq 600$ where $i = 1, 2, \dots, d$.

Ackley’s function is multimodal

$$f(x) = -20 \exp[-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}] - \exp[\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)] + (20 + e) \tag{12}$$

with a global minimum $f_* = 0$ at $x_* = (0,0,\dots,0)$ in the range of $-32.768 \leq x_i \leq 32.768$ where $i = 1, 2, \dots, d$.

The generalized Rosenbrock’s function is given by

$$f(x) = \sum_{i=1}^{d-1} [(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2], \tag{13}$$

which has a minimum $f_* = 0$ at $x_* = (1, 1, \dots, 1)$.

Schwefel’s test function is also multimodal

$$f(x) = \sum_{i=1}^d [-x_i \sin(\sqrt{|x_i|})], \quad -500 \leq x_i \leq 500, \tag{14}$$

with a global minimum of $f_* = -418.9829$ at $x_i^* = 420.9687$

($i = 1, 2, \dots, d$).

Rastrigin’s test function

$$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)], \tag{15}$$

has a global minimum $f_* = 0$ at $x_* = (0,0,\dots,0)$ in a hypercube $-5.12 \leq x_i \leq 5.12$ where $i = 1, 2, \dots, d$.

Michalewicz’s test function has $d!$ local optima

$$f(x) = -\sum_{i=1}^d \sin(x_i) [\sin(\frac{i x_i^2}{\pi})]^{2m}, \quad (m = 10), \tag{16}$$

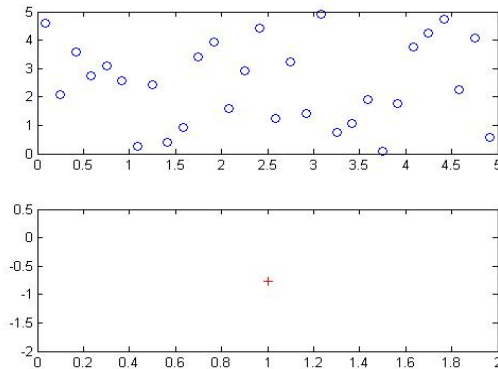


Fig. 2. Initial distribution of cuckoo nest

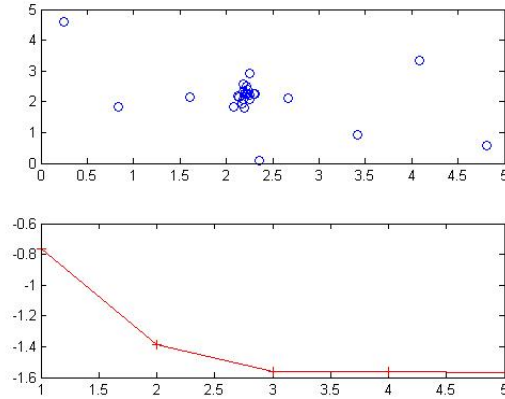


Fig. 3. Position of cuckoo nests and fitness value after 5 generations

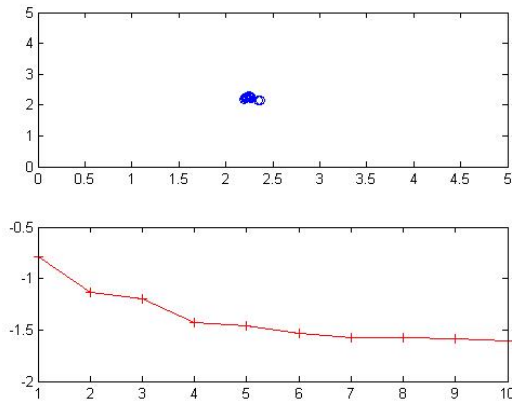


Fig. 4. Position of cuckoo nests and fitness value after 10 generations

4.3 Comparison of ISCS with CS, PSO and GA

After implementing the proposed ISCS algorithm with Matlab we compare it with basic Cuckoo Search (CS), particle swarm optimization (PSO), and genetic algorithms (GA) for various standard test funtions.

We have carried out extensive simulations in Matlab and each algorithm has been run at least 100 times so as to carry out meaningful statistical results. Algorithms were made to stop when the variations of function values are less than a given tolerance $\mathcal{E} \leq 10^{-10}$. The results are summarized in the following tables (see Tables 1 and 2) where the global optima are reached. The numbers are in the format: average number of function evaluations (success rate), so 927 ± 105 (100%) means that the average number (mean) of function evaluations is 927 with a standard deviation of 105. The success rate of finding the global optima for this algorithm is 100%.

From the results, we see that the ISCS algorithm performs much better than PSO, CS and GA for all test functions. Thus, we can say that the ISCS is much more efficient in finding global optimum with higher success rate in both unimodal and multi-modal functions.

| Functions | Function evaluations needed to reach stopping criteria | | | |
|-----------------------|--|--------------------|--------------------|-------------------------|
| | PSO | GA | CS | ISCS |
| Multiple peaks | 3719 ± 205(97%) | 52124 ± 3277(98%) | 927 ± 105(100%) | 907 ±50 (100%) |
| Michalewicz's (d=16) | 6922 ± 537(98%) | 89325 ± 7914(95%) | 3221 ± 519(100%) | 3100±34 (100%) |
| Rosenbrock's (d=16) | 32756 ± 5325(98%) | 55723 ± 8901(90%) | 5923 ± 1937(100%) | 5632±776 (100%) |
| De Jong's (d=256) | 17040 ± 1123(100%) | 25412 ± 1237(100%) | 4971 ± 754(100%) | 4264±606 (100%) |
| Schwefel's (d=128) | 14522 ± 1275(97%) | 227329± 7572(95%) | 8829 ± 625(100%) | 7665±241 (100%) |
| Ackley's (d=128) | 23407 ± 4325(92%) | 32720 ± 3327(90%) | 4936 ± 903(100%) | 4115±586 (100%) |
| Rastrigin's | 79491 ± 3715(90%) | 110523± 5199(77%) | 10354 ± 3755(100%) | 9753±1709 (100%) |
| Easom's | 17273 ± 2929(90%) | 19239 ± 3307(92%) | 6751 ± 1902(100%) | 5109±1397 (100%) |
| Griewank's | 55970 ± 4223(92%) | 70925 ± 7652(90%) | 10912 ± 4050(100%) | 9122±4021 (100%) |
| Shubert's (18 minima) | 23992 ± 3755(92%) | 54077 ± 4997(89%) | 9770 ± 3592(100%) | 9003±3209 (100%) |

5 Conclusion

In this paper we have proposed an improved version of Cuckoo Search (CS). Inter-species Cuckoo Search (ISCS) is a new meta-heuristic Cuckoo Search based on the breeding strategy of different cuckoo species living in different regions. The Levy flights performed by different species of cuckoo are also different in nature. The proposed algorithm uses clustering to divide different cuckoo species and there is sufficient information sharing between different cuckoo species. The Matlab simulations and comparisons validate that the proposed algorithm outperforms other algorithms such as genetic algorithms, particle swarm optimization and basic Cuckoo Search in both unimodal and multi-modal objective functions. The reasons behind such success are the existence of different cuckoo species as well as different levy flights and reduction in the number of parameters to be adjusted. In fact, the only parameter to be fine-tuned was the population size n , which was fixed for all functions and it was found that the convergence rate was almost insensitive to this parameter. The other parameter used in basic Cuckoo Search (CS) was the probability, p_a but in

case of our proposed algorithm every cluster or region has a randomized value of p_a and thus we do not need any parameter adjustment for different functions.

The Inter-Species Cuckoo Search was applied on many other real world problems such as Economic load dispatch (ELD) problems of various range, some power system related problems and some other engineering problems such as permutation flowshop scheduling problem, hybrid flowshop scheduling problem and railway time-table scheduling problems. The last three problems have discrete solution representation; hence the ISCS was modified accordingly to be suitable for application in discrete domain. The ISCS was also applied in CEC 2011 benchmark problems [28] where its performance was compared with that of PSO, GA, DE and some other meta-heuristics. The results were very promising but due to lack of sufficient space we are unable to provide all the tables and necessary data. In future, we will surely publish the complete set of results and comparisons.

References

- [1] Yang, X.-S., Deb, S.: Engineering Optimisation by Cuckoo Search. *Int. J. Mathematical Modelling and Numerical Optimisation* 1(4), 330–343 (2010)
- [2] Yang, X.S., Deb, S.: Engineering Optimisation by Cuckoo Search. *Int. J. of Mathematical Modelling and Numerical Optimisation* 1(4), 330–343 (2010)
- [3] Yang, X.-S.: A New Metaheuristic Bat-Inspired Algorithm. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (eds.) *NICSO 2010. SCI*, vol. 284, pp. 65–74. Springer, Heidelberg (2010)
- [4] Bonabeau, E., Dorigo, M., Theraulaz, G.: *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press (1999)
- [5] Deb, K.: *Optimisation for Engineering Design*. Prentice-Hall, New Delhi (1995)
- [6] Yang, X.S.: *Nature-Inspired Metaheuristic Algorithms*. Luniver Press (2008)
- [7] Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.* 35, 268–308 (2003)
- [8] Yang, X.S.: Biology-derived algorithms in engineering optimization. In: Olariu, Zomaya (eds.) *Handbook of Bioinspired Algorithms and Applications*, ch.32. Chapman & Hall / CRC
- [9] Kennedy, J., Eberhart, R., Shi, Y.: *Swarm intelligence*. Academic Press (2001)
- [10] Payne, R.B., Sorenson, M.D., Klitz, K.: *The Cuckoos*. Oxford University Press (2005)
- [11] Brown, C., Liebovitch, L.S., Glendon, R.: Lévy flights in Dobe Ju/'hoansi foraging patterns. *Human Ecol.* 35, 129–138 (2007)
- [12] Pavlyukevich, I.: Lévy flights, non-local search and simulated annealing. *J. Computational Physics* 226, 1830–1844 (2007)
- [13] Pavlyukevich, I.: Cooling down Lévy flights. *J. Phys. A: Math. Theor.* 40, 12299–12313 (2007)
- [14] Reynolds, A.M., Frye, M.A.: Free-flight odor tracking in *Drosophila* is consistent with an optimal intermittent scale-free search. *PLoS One* 2, e354 (2007)
- [15] Shlesinger, M.F., Zaslavsky, G.M., Frisch, U. (eds.): *Lévy Flights and Related Topics in Physics*. Springer (1995)
- [16] Shlesinger, M.F.: Search research. *Nature* 443, 281–282 (2006)
- [17] Chattopadhyay, R.: A study of test functions for optimization algorithms. *J. Opt. Theory Appl.* 8, 231–236 (1971)

- [18] Schoen, F.: A wide class of test functions for global optimization. *J. Global Optimization* 3, 133–137 (1993)
- [19] Shang, Y.W., Qiu, Y.H.: A note on the extended rosenbrock function. *Evolutionary Computation* 14, 119–126 (2006)
- [20] Hartigan, J.A., Wong, M.A.: Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society, Series C (Applied Statistics)* 28(1), 100–108 (1979)
- [21] Ding, C., He, X.: K-means Clustering via Principal Component Analysis. In: *Proc. of Int'l Conf. Machine Learning (ICML)*, pp. 225–232 (July 2004)
- [22] Winkel, M.: Introduction to Lévy processes. pp. 15–16 (retrieved January 07, 2013)
- [23] Alamgir, M., von Luxburg, U.: Multi-agent random walks for local clustering on graphs. In: *IEEE 10th International Conference on Data Mining (ICDM)*, pp. 18–27 (2010)
- [24] Sijbers, J., den Dekker, A.J., Raman, E., Van Dyck, D.: Parameter estimation from magnitude MR images. *International Journal of Imaging Systems and Technology* 10(2), 109–114 (1999)
- [25] Srivastava, P.R., Varshney, A., Nama, P., Yang, X.S.: Software test effort estimation: a model based on cuckoo search. *International Journal of Bio-inspired Computation* 4(5), 278–285 (2012)
- [26] Marichelvam, M.K.: An improved hybrid Cuckoo Search (IHCS) metaheuristics algorithm for permutation flow shop scheduling problems. *International Journal of Bio-inspired Computation* 4(4), 200–205 (2012)
- [27] Gherboudj, A., Layeb, A., Chikhi, S.: Solving 0-1 knapsack problems by a discrete binary version of cuckoo search algorithm. *International Journal of Bio-inspired Computation* 4(4), 229–236 (2012)
- [28] Swagatam, D., Suganthan, P.N.: Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems. Technical Report (December 2010)

Cuckoo Search Algorithm for the Mobile Robot Navigation

Prases Kumar Mohanty and Dayal R. Parhi

Robotics Laboratory, Department of Mechanical Engineering ,
National Institute of Technology, Rourkela
Odisha, India
pkmohanty30@gmail.com, dayalparhi@yahoo.com

Abstract. The shortest/optimal path planning is essential for efficient operation of autonomous vehicle. In this paper a cuckoo search based approach has been implemented for mobile robot navigation in an unknown environment populated by a variety of obstacles. This metaheuristic algorithm is based on the levy flight behavior and brood parasitic behavior of cuckoos. A new objective function has been developed between robot and position of the goal and obstacles present in the environment. Depending upon the objective function value of each nest in swarm, the robot avoids obstacles and proceeds towards goal. The optimal path is generated with this algorithm when robot reaches its goal. Several simulation results are presented here to demonstrate the potential of proposed algorithm.

Keywords: Cuckoo search, Levy flight, Mobile robot, path planning.

1 Introduction

Autonomous path planning of mobile robot has acquired considerable attention in recent years. The major issue in autonomous mobile robot is its path planning in uncertain and complex environment. If robot wants to travel among the unknown obstacles to reach a specified goal without collisions, then various sensors must be needed to perceive information about the real world environment. The sensor based motion planning approaches uses either global or local path planning depending upon the surrounding environment. Global path planning requires the environment to be completely known and the terrain should be static, on other side local path planning means the environment is completely or partially unknown for the mobile robot. Many exertions have been paid in the past to improve various robot navigation techniques.

In literature review, there can be found several researchers have been worked on many intelligent techniques for mobile robot navigation. Many authors have considered a controller with complete information of the environment [1-2]. Due to the complexity and uncertainty of the path planning problem, classical path planning methods, such as visibility Graph [3], voronoi diagrams [4], grids [5], cell decomposition [6], artificial potential field [7], rule Based Methods [8], and rules learning techniques [9] are not appropriate for path planning in dynamic

environments. The use of the above algorithms for path finding for mobile robot requires more time and the finding of this path will not completely feasible for real-time movement. In recent times most popular metaheuristic algorithms such as genetic algorithm (GA) [10-12], simulated annealing (SA) [13-14], ant colony optimization (ACO) [15-16], firefly algorithm (FA) [17], and particle swarm optimization (PSO) [18-20] have been implemented for path planning of mobile robot. Most of these metaheuristic algorithms are inspired, mimicking the successful features of the underlying biological, physical or sociological systems.

In this paper, we introduce a new variant of cuckoo search algorithm for solving the path planning problem of mobile robot. Cuckoo search is an optimization metaheuristic search algorithm developed by Yang and Deb in 2009[21-22], which was inspired by obligate parasitic behavior of some cuckoo, by laying their eggs in nest of other host birds. An advantage of CS is that, the number of parameters to be tuned is less than GA and PSO and thus it is potentially more generic to implement to a wider class of optimization problem. Finally, simulation results are presented to verify the effectiveness of the proposed algorithm in various scenarios populated by variety of static obstacles.

2 Cuckoo Search Algorithm

In nature, cuckoo has an aggressive reproduction strategy that involves the female laying her fertilized eggs in the nest of another species to let host birds to hatch and brood young cuckoo chicks. Sometimes the host bird discover that the eggs are not it's own and either destroy the cuckoo egg or abandon the nest and start their own brood elsewhere.

Cuckoo search is a newly developed metaheuristic algorithm for solving optimization problems, which based on the obligate brood parasitic behavior of some cuckoo species in combination with the levy flight behavior of some birds. In CS algorithm, levy flight is used for generating nests or solutions after each iteration.

A levy flight is type of random walk in which steps are distributed in terms of step lengths according to a heavy tailed probability distribution. Levy flight is a random walk in which the increments are distributed according to power law that is

$$y = x^{-\beta}$$

where $1 < \beta < 3$ and therefore has an infinite variance.

For simplicity in describing new cuckoo search algorithm [21], Yang and Deb used the following three idealized rules

- 1) Each cuckoo lays one egg at a time, and dump its egg in randomly chosen nest;
- 2) The best nests with high quality of eggs will carry over to the next generations;
- 3) The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in [0,1]$.

In this case, the host bird can either throw the egg away or abandon the nest, and build a completely new nest at new location. For simplicity, this last assumption can be

approximated by the fraction p_a of the n nests are replaced by new nests (with new random solutions).

For a maximization problem, the quality or fitness of a solution can simply be proportional to the value of the objective function. Other forms of fitness can be defined in a similar way to the fitness function in genetic algorithms. For simplicity, we can use the following simple representations that each egg in a nest represents a solution, and a cuckoo egg represent a new solution, the aim is to use the new and potentially better solutions (cuckoos) to replace a not-so-good solution in the nests. Of course, this algorithm can be extended to the more complicated case where each nest has multiple eggs representing a set of solutions. For this present problem, we will use the simplest approach where each nest has only a single egg.

Based on these three rules, the basic steps of the Cuckoo Search (CS) can be summarized as the pseudo code shown in Fig. 1.

When generating new solutions $X^{(t+1)}$ for, say, a cuckoo i , a levy flight is performed

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus \text{levy}(\lambda)$$

where $\alpha > 0$ is the step size which should be related to the scales of the problem of interests. In most cases, we can use $\alpha = 1$. The above equation is essentially the stochastic equation for random walk. In general, a random walk is a Markov chain whose next status/location only depends on the current location (the first term in the above equation) and the transition probability (the second term). The product \oplus means entry wise multiplications. This entry wise product is similar to those used in PSO, but here the random walk via Levy flight is more efficient in exploring the search space as its step length is much longer in the long run.

```

Objective function  $f(x)$ ,  $x = (x_1, x_2, \dots, x_d)^T$ 
Generate initial population of  $n$  host nests  $x_i (i = 1, 2, \dots, n)$ 
while ( $t < \text{Max Generation}$ ) or (Stop criterion) do
  Get a cuckoo randomly by Levy flights
  Evaluate its quality/fitness  $F_i$ 
  Choose a nest among  $n$  (say  $j$ ) randomly
  if ( $F_i > F_j$ ) then
    replace  $j$  by the new solution
  end
  A fraction ( $p_a$ ) of worse nests are abandoned and new ones are built
  Keep the best solutions (or nests with quality solutions)
  Rank the solutions and find the current best
End while
Post process results and visualization
End

```

Fig. 1. Algorithm of Cuckoo search

3 Architecture of Robot Path Planning with Cuckoo Search Technique

Cuckoo search is a new optimization concept which broadly falls under swarm evolutionary computation techniques. The objective considered in the section is robot path planning in an unknown environment populated by variety of obstacles. Based on the sensory information about the target and obstacles, we adopt cuckoo search algorithm to solve it. We firstly transform the navigation problem into a minimization one and formulate a objective function equation based on the position of the target and the obstacles present in the environment. Then we implement CS to solve the above optimization problem. During this process results, the locations of the globally best nest in each iterative are selected and robot reaches these locations in sequence. When the robot does not detect any obstacles in its target path, then it will travel towards its destination. Then it is not necessary to implement any intelligence technique to travel the robot within its environment. The developed flow chart diagram for the current analysis is given in Fig [2].

Formulation of objective function

Each solution in the problem space is associated with a numeric value. In CS, a nest egg of best quality will lead to a new generation. Therefore the quality of a cuckoo's egg (new cuckoo) is related to the optimized path length for the robot. As is explained in the above optimization algorithm, each location to be reached by the robot is calculated based on the distance between itself and the goal and obstacles present in the environment. In general, the robot path planning is mainly depends upon the following two conditions,

- a) Robot should have kept up maximum distance from the nearest obstacle. The distance between the robot and obstacle is calculated by the following equation,

$$(\text{Dist})_{\text{ROB}} = \sqrt{(x_{\text{OB}_N} - x_{\text{ROB}})^2 + (y_{\text{OB}_N} - y_{\text{ROB}})^2}$$

- b) Robot should have kept up minimum distance from the goal. The distance between the robot and goal is calculated by the following equation,

$$(\text{Dist})_{\text{RG}} = \sqrt{(x_G - x_{\text{ROB}})^2 + (y_G - y_{\text{ROB}})^2}$$

Based on the above two conditions the objective function of the nest can be expressed as follows:

$$f(n_i) = c_1 \cdot \frac{1}{\min \|n_i - \text{OB}_j\|} + c_2 \cdot \|n_i - G\|$$

$$\text{OB}_j \in \text{OB}_d$$

Based on the above objective function, we notate G as the goal, whose coordinate is (x_G, y_G) . In addition we assume that the 'N' number of obstacles present in the environment and denote them as $\text{OB}_1, \text{OB}_2, \text{OB}_3 \dots \text{OB}_N$, their coordinates are $(x_{\text{OB}_1}, y_{\text{OB}_1}), (x_{\text{OB}_2}, y_{\text{OB}_2}), (x_{\text{OB}_3}, y_{\text{OB}_3}) \dots (x_{\text{OB}_N}, y_{\text{OB}_N})$. Due to limit range of the robot sensor, in each step it can able to recognize a number of obstacles present in the environment and number of obstacles being recognized by the robot sensor in some step is denoted as $\text{OB}_d \in \{ \text{OB}_1, \text{OB}_2, \text{OB}_3, \dots \text{OB}_N \}$.

It can be seen from the objective function that when n_i is nearer to the goal, the value of $\|n_i - G\|$ will be less and when n_i is far from the obstacles, the value of $\min_{OB_j \in OB_d} \|n_i - OB_j\|$ will be large. So the problem solved by CS is a minimization one.

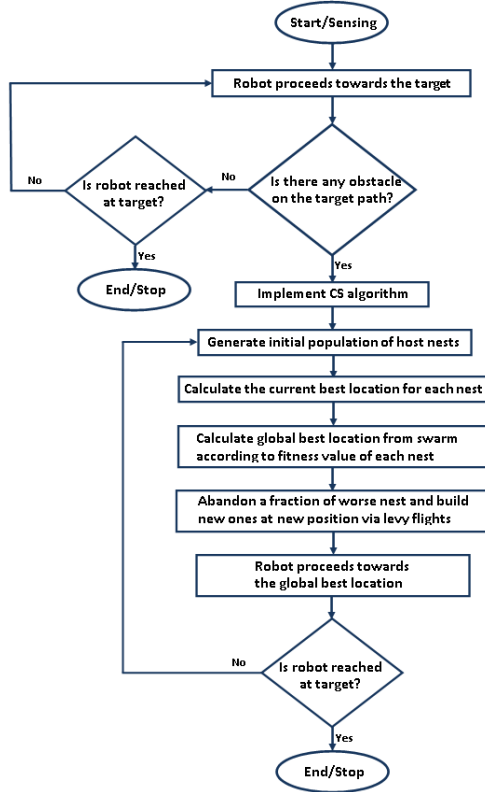


Fig. 2. Flow chart represented for the proposed algorithm

From the objective function equation, it can be clearly understood that the controlling or fitness parameters c_1 and c_2 have influence on the robot trajectory. When c_1 is large, the robot will be far away from the obstacles, if it is less there will be chance to collide with them. Other side, when c_2 is large, robot has a strong potential to move to goal, resulting the path length being short, otherwise it will be large.

4 Simulation Results

We have tested our proposed algorithm in two dimensional path planning through series of simulation experiments under unknown or partially known environment. Our implementation was compiled using MATLAB R2008a processing under Windows XP. All the simulation results were applied on PC with Intel core2 processor running at 3.0GHz, 4GB of RAM and a hard drive of 160GB.

As per the simulation experiments by Yang and Deb [21], it was found that population size $n=15$ to 40 and discovery rate $pa=0.25$ are suitable for solving most of optimization problems. In order to start the proposed algorithm for path planning of mobile robot, the following parameters of CS are taken into consideration.

No. of Nest (population size) $n=30$

Step size $\alpha=1$

Probability of discovery rate $=0.25$

Maximum generation $=300$ nos.

The simulation results were also compared with GA and PSO and it is verified that using proposed algorithm the robot reached to the specified target in optimum path length (Table-1).

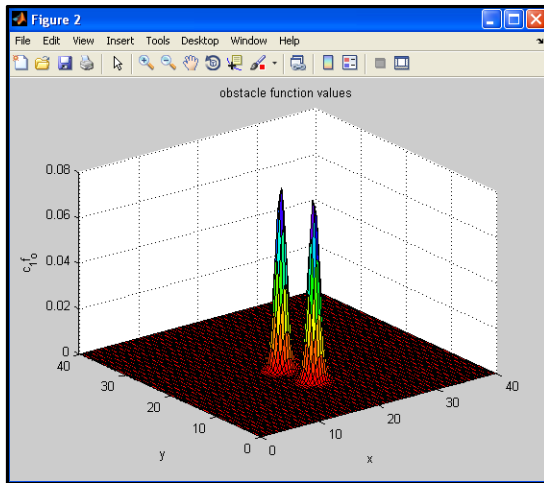


Fig. 3. Fitness landscape of obstacle function

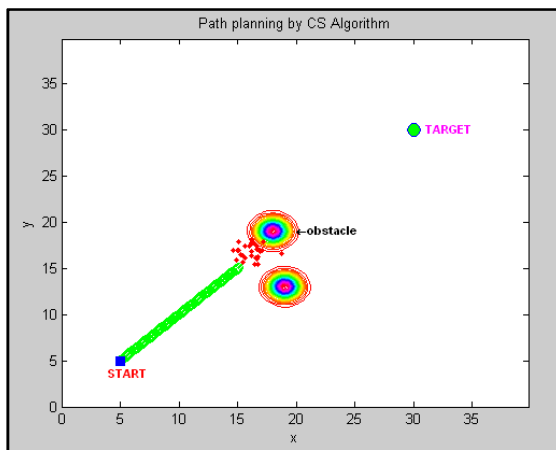


Fig. 4. Activation of CS algorithm for obstacle avoidance

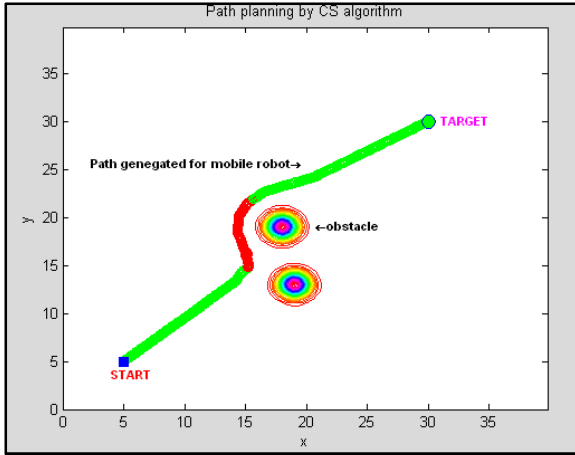


Fig. 5. Obstacle avoidance by single robot using current algorithm ($C1=1, C2=1.0e-04$)

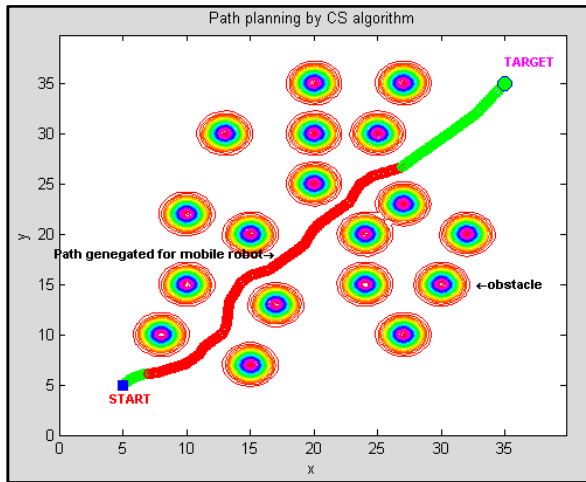


Fig. 6. Obstacle avoidance by single robot in maze environment

Table 1.

| SL No. | Path length cover by the robot in pixels | | |
|--------|--|-------|-------|
| | Fig.5 | Fig.6 | Fig.7 |
| CS | 170 | 310 | 138 |
| PSO | 198 | 329 | 152 |
| GA | 203 | 341 | 166 |

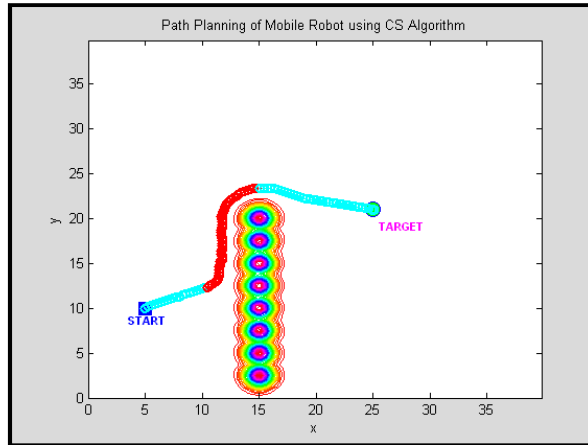


Fig. 7. Wall following by single robot using current algorithm
($C1=0.8$, $C2=1.0e-03$)

5 Conclusion and Future Work

In this research paper, a new metaheuristic algorithm has been developed for path planning problem of autonomous mobile robot in an unknown or partially known environment populated by variety of static obstacles. It has been found that the CS algorithm is capable of avoiding obstacles and effectively guiding the mobile robot moving from the start point to the desired destination point with optimum/shortest path length. The authenticity of the proposed algorithm has been verified and proven by simulation experiments using MATLAB. The simulation results were also compared with GA and PSO and it is verified that using proposed algorithm the robot reached to the specified target in optimum path length. In future real time implementation is to be carried out using robot and multiple robots are to be considered instead of a single mobile robot.

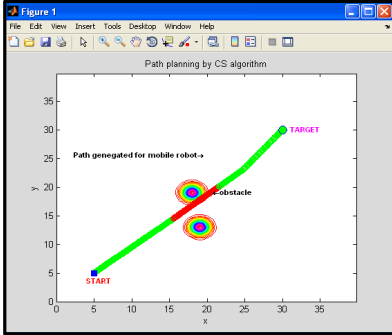
References

1. Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, New York (1990)
2. Canny, J.E.: The Complexity of Robot Motion Planning. MIT Press, Cambridge (1988)
3. Lozano-Perez, T.: A simple motion planning algorithm for general robot manipulators. IEEE Journal of Robotics and Automation 3, 224–238 (1987)
4. Leven, D., Sharir, M.: Planning a purely translational motion for a convex object in two dimensional space using generalized voronoi diagrams. Discrete & Computational Geometry 2, 9–31 (1987)
5. Payton, D., Rosenblatt, J., Keirse, D.: Grid-based mapping for autonomous mobile robot. Robotics and Autonomous Systems 11, 13–21 (1993)

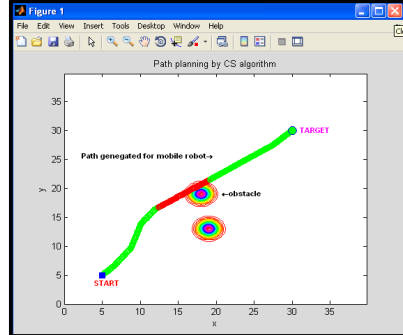
6. Regli, L.: Robot Lab: Robot Path Planning. Lectures Notes of Department of computer Science. Drexel University (2007)
7. Khatib, O.: Real time Obstacle Avoidance for manipulators and Mobile Robots. In: IEEE Conference on Robotics and Automation, vol. 2, pp. 505–505 (1985)
8. Fujimura, K.: Motion Planning in Neritic Environments. Springer (1991)
9. Ibrahim, M.Y., Fernandes, A.: Study on Mobile Robot Navigation Techniques. In: IEEE International Conference on Industrial Technology, vol. 1, pp. 230–236 (2004)
10. Castillo, O., Trujillo, L., Melin, P.: Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots. *Soft Computing* 11, 269–279 (2007)
11. Hamdan, M., El-Hawary, M.E.: A novel genetic algorithm searching approach for dynamic constrained multicast routing. In: Proc. IEEE/CCECE, pp. 1127–1130 (2003)
12. Zein-Sabatto, S., Ramakrishnan, R.: Multiple path planning for a group of mobile robots in a 3D environment using genetic algorithms. In: Proc. IEEE Southeast, pp. 359–363 (2002)
13. Qidan, Z., Yongjie, Y., Zhuoyi, X.: Robot Path Planning Based on Artificial Potential Field Approach with Simulated Annealing. In: Proc. ISDA 2006, pp. 622–627 (2006)
14. Park, M.G., Lee, M.C.: Experimental evaluation of robot path planning by artificial potential field approach with simulated annealing. In: Proc. SICE, vol. 4, pp. 2190–2195 (2002)
15. Chen, X., Kong, Y., Fang, X., Wu, Q.: A fast two-stage ACO algorithm for robotic path planning. *Neural Computing and Applications* 22(2), 313–319 (2013)
16. Shirong, L., Linbo, M., Jinshou, Y.: Path Planning Based on Ant Colony Algorithm and Distributed Local Navigation for Multi-Robot Systems. In: Proc. IEEE Int. Conf. on Mechatronics and Automation, pp. 1733–1738 (2006)
17. Liu, C., Gao, Z., Zhao, W.: A New Path Planning Method Based on Firefly Algorithm. In: Fifth International Joint Conference on Computational Sciences and Optimization, pp. 775–778 (2012)
18. Zhang, Y., Gong, D.W., Zhang, J.H.: Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neuro-Computing* 103, 172–185 (2013)
19. Tang, Q., Eberhard, P.: A PSO-based algorithm designed for a swarm of mobile robots. *Structural and Multidisciplinary Optimization* 44, 483–498 (2011)
20. Saska, M., Macas, M., Preucil, L., Lhotska, L.: Robot Path Planning using Particle Swarm Optimization of Ferguson Splines. In: Proc. IEEE/ETFA, pp. 833–839 (2006)
21. Yang, X.S., Deb, S.: Cuckoo Search via Levy Flights. In: Proc. of World Congress on Nature & Biologically Inspired Computing, pp. 210–214 (2009)
22. Deb, S.: Engineering Optimisation by Cuckoo Search. *International Journal of Mathematical Modelling and Numerical Optimisation* 3, 330–343 (2010)
23. Yang, X.S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press (2010)
24. Yang, X.S., Deb, S.: Cuckoo search: recent advances and applications. *Neural Computing and Application* (2013), doi:10.1007/s00521-013-1367-1
25. Yang, X.S., Deb, S.: Discrete Cuckoo search algorithm for the travelling salesman problem. *Neural Computing and Application* (2013), doi:10.1007/s00521-013-1402-2
26. Rakshit, P., Konar, A., Bhowmik, P., Goswami, I., Das, S., Jain, L.C., Nagar, A.K.: Realization of an Adaptive Memetic Algorithm Using Differential Evolution and Q-Learning: A Case Study in Multi-robot Path Planning. *IEEE T. Systems, Man, and Cybernetics: Systems* 43(4), 814–831 (2013)
27. Panda, R., Agrawal, S., Bhuyan, S.: Edge Magnitude based Multilevel Thresholding using Cuckoo Search Technique. *Expert Systems with Applications* 40(18), 7617–7628 (2013)

Appendix I: Tuning of Parameters C1 and C2

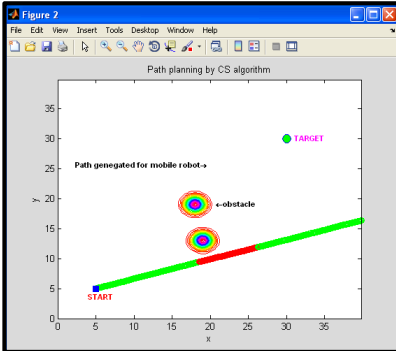
The choice of the controlling parameters (C1,C2) will affect the shape of the trajectory that the robot will move on toward the target position.



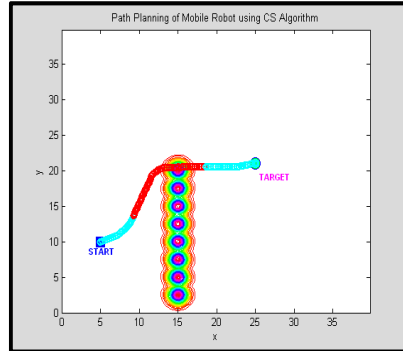
C1=2, C2=1.0e-05



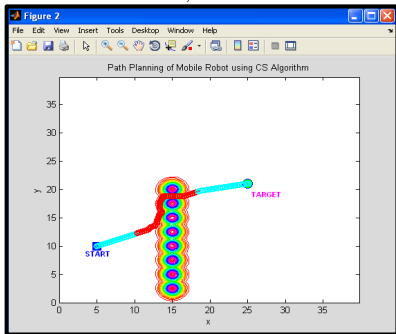
C1=3, C2=1.0e-07



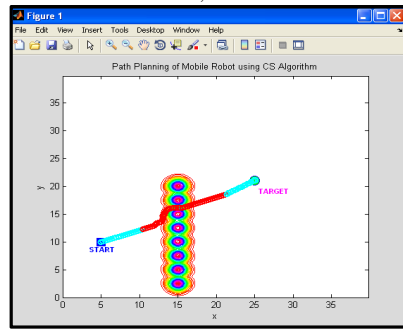
C1=1, C2=1.0e-10



C1=2.8, C2=1.0e-09



C1=3.4, C2=1.0e-09



C1=5, C2=1.0e-08

Automatic Generation Control of Multi-area Power System Using Gravitational Search Algorithm

Rabindra Kumar Sahu*,
Umesh Kumar Rout, and Sidhartha Panda

Department of Electrical Engineering
Veer Surendra Sai University of Technology (VSSUT), Burla-768018, Odisha, India
rksahu123@gmail.com, umesh6400@gmail.com,
panda_sidhartha@rediffmail.com

Abstract. This paper presents the design and performance analysis of Gravitational Search algorithm (GSA) based Proportional Integral (PI) controller and Proportional Integral Derivative controller with derivative Filter (PIDF) for Automatic Generation Control (AGC) of multi-area power system. At first, various conventional error criterions such as Integral of Absolute Error (IAE), Integral of Square Error (ISE), Integral of Time multiplied by Square Error (ITSE) and Integral of Time multiplied by Absolute value of Error (ITAE) are considered and the PI controller parameters are optimized employing GSA. The effect of objective function on system performance in terms of settling time, maximum overshoot and minimum damping ratio are analyzed. The control parameters of GSA algorithm are tuned by carrying out multiple runs of algorithm for each control parameter variation. The superiority of the proposed GSA optimized PI/PIDF controller is demonstrated by comparing the results with some recently published modern heuristic optimization techniques such as Differential Evolution (DE), Bacteria Foraging Optimization Algorithm (BFOA) and Genetic Algorithm (GA) for the same interconnected power system.

Keywords: Automatic Generation Control (AGC), Proportional-Integral (PI) controller, Proportional Integral Derivative controller with derivative Filter (PIDF), Gravitational Search Algorithm (GSA).

1 Introduction

The main objective of a power system utility is to maintain continuous supply of power with an acceptable quality to all the consumers in the system. The system will be in equilibrium, when there is a balance between the power demand and the power generated [1],[12]. There are two basic control mechanisms used to achieve power balance; reactive power balance (acceptable voltage profile) and real power balance (acceptable frequency values). The former is called the Automatic Voltage Regulator

* Corresponding author.

(AVR) and the latter is called the Automatic Load Frequency Control (ALFC) or Automatic Generation Control (AGC). For multi area power systems, which normally consist of interconnected control area, AGC is an important aspect to keep the system frequency and the interconnected area tie-line power as close as possible to the intended values [2]. The mechanical input power to the generators is used to control the system as it is affected by the output electrical power demand and to maintain the power exchange between the areas as planned. AGC monitors the system frequency and tie-line flows, calculates the net change in the generation required according to the change in demand and changes the set position of the generators within the area so as to keep the time average of the ACE (Area Control Error) at a low value. ACE is generally treated as controlled output of AGC. As the ACE is adjusted to zero by the AGC, both frequency and tie-line power errors will become zero [3], [4], [11].

The aim of the present work is:

- ❖ to study the effect of objective function of the system performance
- ❖ to tune the control parameters of GSA
- ❖ to demonstrate the advantages of GSA over other techniques such as DE, BFOA and GA which are recently reported in the literature for the similar problem
- ❖ to show advantages of using a modified controller structure and objective function to further increase the performance of the power system

2 System Under Study

System under investigation consists of two area interconnected power system of non reheat thermal plant as shown in Fig. 1. Each area has a rating of 2000 MW with a nominal load of 1000 MW. The system is widely used in literature is for the design and analysis of automatic load frequency control of interconnected areas [6], [7]. In Fig. 1, B_1 and B_2 are the frequency bias parameters; ACE_1 and ACE_2 are area control errors; u_1 and u_2 are the control outputs from the controller; R_1 and R_2 are the governor speed regulation parameters in pu Hz; T_{G1} and T_{G2} are the speed governor time constants in sec; ΔP_{V1} and ΔP_{V2} are the change in governor valve positions (pu); ΔP_{G1} and ΔP_{G2} are the governor output command (pu); T_{T1} and T_{T2} are the turbine time constant in sec; ΔP_{T1} and ΔP_{T2} are the change in turbine output powers; ΔP_{D1} and ΔP_{D2} are the load demand changes; ΔP_{Tie} is the incremental change in tie line power (p.u); K_{PS1} and K_{PS2} are the power system gains; T_{PS1} and T_{PS2} are the power system time constant in sec; T_{12} is the synchronizing coefficient and ΔF_1 and ΔF_2 are the system frequency deviations in Hz. The relevant parameters are taken from [6]. Each area of the power system consists of speed governing system, turbine and generator as shown in Fig.1.

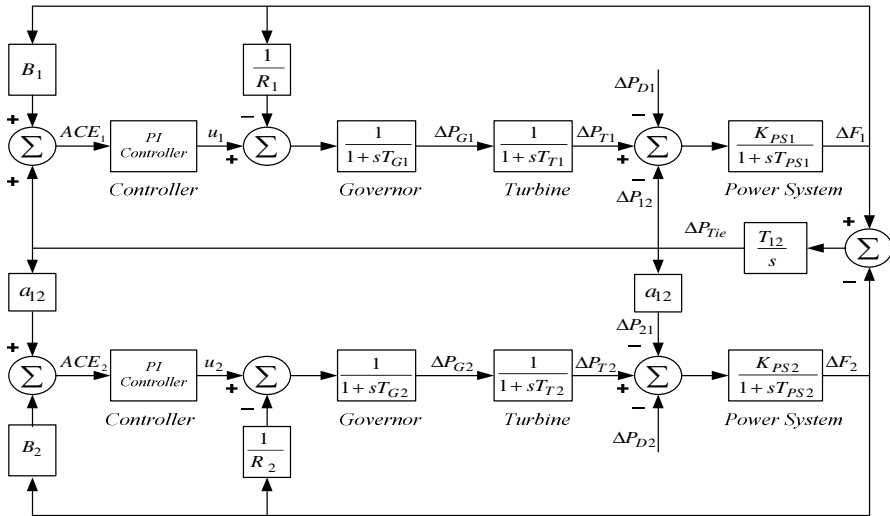


Fig. 1. Transfer function model of two-area non reheat thermal system

3 Overview of Gravitational Search Algorithm

Gravitational Search Algorithm (GSA) is one of the newest heuristic algorithms inspired by the Newtonian laws of gravity and motion [8]. In GSA, agents are considered as objects and their performance is measured by their masses. All these objects attract each other by the force of gravity and this force causes a global movement of all objects towards the objects with a heavier mass. Hence masses cooperate using a direct form of communication through gravitational force.

3.1 Law of Gravity

Each particle attracts every other particle and the gravitational force between the two particle is directly proportional to the product of their masses and inversely proportional to the distance between them R. It has been reported in literature that R provides better results than R² in all experiment cases [9].

3.2 Law of Motion

The current velocity of any mass is equal the sum of the fraction of its previous velocity and the variation in the velocity. Variation in the velocity or acceleration of any mass is equal to the force acted on the system divided by mass of inertia.

For a system with ‘n’ agent (masses), the *i*-th position of an agent *X_i* is defined by:

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \tag{1}$$

for *i* = 1, 2, ..., *n*

Where

x_i^d represents the position of i -th agent in the d -th dimension.

At a specific time ' t ', the force acting on mass ' i ' from mass ' j ' is defined as:

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) * M_{aj}(t)}{R_{ij}(t) + \epsilon} (x_j^d(t) - x_i^d(t)) \quad (2)$$

Where

M_{aj} is the active gravitational mass related to agent j

M_{pi} is the passive gravitational mass related to agent i

$G(t)$ is the gravitational constant at time t

ϵ is small constant

$R_{ij}(t)$ is the Euclidian distance between two agents i and j given by:

$$R_{ij}(t) = \|X_i(t), X_j(t)\|_2 \quad (3)$$

The stochastic characteristic in GSA algorithm is incorporated by assuming that the total forces that act on agent ' i ' in a dimension ' d ' be a randomly weight sum of d -th components of the forces exerted from other agents as:

$$F_i^d(t) = \sum_{j=1, j \neq i}^n rand_j F_{ij}^d(t) \quad (4)$$

Where $rand_j$ is a random number in the interval $[0, 1]$

The acceleration of the agent ' i ' at the time t and in the direction d -th, $a_i^d(t)$ is given by the law of the motion as:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (5)$$

Where $M_{ii}(t)$ is the inertia mass of i -th agent.

The velocity of an agent is updated depending on the current velocity and acceleration. The velocity and position are updated as:

$$v_i^d(t+1) = rand_i * v_i^d(t) + a_i^d(t) \quad (6)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (7)$$

Where $rand_i$ is a uniform random variable in the interval $(0, 1)$. The random number is used to give a randomized characteristic to the search process.

The gravitational constant G is initialized at the beginning. To control the search accuracy it is reduced with time and expressed as function of the initial value (G_0) and time t as:

$$G(t) = G_0 e^{(-\alpha t/T)} \quad (8)$$

Where α is a constant and T is the number of iteration.

To achieve a good compromise between exploration and exploitation, the number of agents is reduced with lapse of Eq. (5) and therefore a set of agents with bigger mass are used for applying their force to the other. The performance of GSA is improved by controlling exploration and exploitation. To avoid trapping in a local optimum GSA must use the exploration at beginning. By lapse of iterations, exploration must fade out and exploitation must fade in.

4 Proposed Approach

4.1 Controller Structurer

In the present paper, identical controllers have been considered for the two areas as the two areas are identical. The structure of PID controller with derivative filter is shown in Fig. 2 where K_P , K_I and K_D are the proportional, integral and derivative gains respectively, and N is the derivative filter coefficient. When used as PI controller, the derivative path along with the filter is removed from Fig. 2.

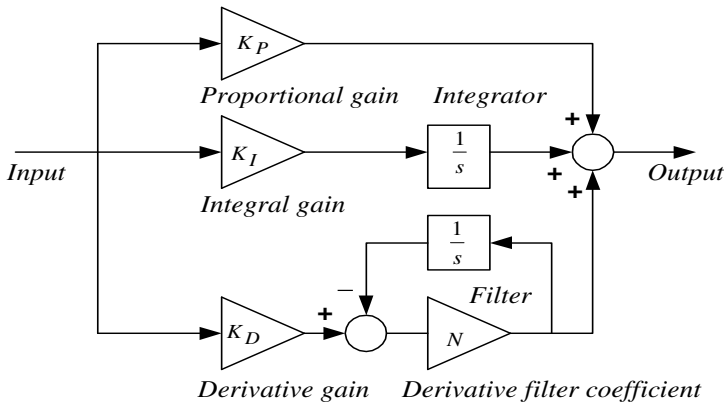


Fig. 2. Structure of PID controller with derivative filter

4.2 Objective Function

To determining the optimum values of controller parameters conventional objective functions are considered at the first instance. The objective functions considered are IAE, ISE, ITSE and ITAE as given below:

$$J_1 = ISE = \int_0^{t_{sim}} (|\Delta F_1| + |\Delta F_2| + |\Delta P_{Tie}|)^2 .dt \quad (9)$$

$$J_2 = IAE = \int_0^{t_{sim}} (|\Delta F_1| + |\Delta F_2| + |\Delta P_{Tie}|) .dt \quad (10)$$

$$J_3 = ITSE = \int_0^{t_{sim}} \left(|\Delta F_1| + |\Delta F_2| + |\Delta P_{Tie}| \right)^2 \cdot t \cdot dt \quad (11)$$

$$J_4 = ITAE = \int_0^{t_{sim}} \left(|\Delta F_1| + |\Delta F_2| + |\Delta P_{Tie}| \right) \cdot t \cdot dt \quad (12)$$

$$J_5 = \int_0^{t_{sim}} \omega_1 \cdot \int_0^{t_{sim}} \left(|\Delta F_1| + |\Delta F_2| + |\Delta P_{Tie}| \right) \cdot t \cdot dt + \omega_2 \cdot \frac{1}{\min\left(\sum_{i=1}^n (1 - \zeta_i)\right)} + \omega_3 (ST) \quad (13)$$

Where, ΔF_1 and ΔF_2 are the system frequency deviations; ΔP_{Tie} is the incremental change in tie line power; t_{sim} is the time range of simulation; ζ_i is the damping ratio and n is the total number of the dominant eigen values; ST is the sum of the settling times of frequency and tie line power deviations respectively; ω_1 to ω_3 are weighting factors. The problem constraints are the PI/PIDF controller parameter bounds. Therefore, the design problem can be formulated as the following optimization problem.

$$\text{Minimize } J \quad (13)$$

Subject to

$$\text{For PI controller: } K_{P \min} \leq K_P \leq K_{P \max}, K_{I \min} \leq K_I \leq K_{I \max} \quad (14)$$

For PIDF controller:

$$K_{P \min} \leq K_P \leq K_{P \max}, K_{I \min} \leq K_I \leq K_{I \max}, K_{D \min} \leq K_D \leq K_{D \max} \quad (15)$$

The minimum and maximum values of PID controller parameters are chosen as -2.0 and 2.0 respectively. The range for filter coefficient N is selected as 1 and 300.

5 Results and Discussions

5.1 Application of GSA

The model of the system under study is developed in MATLAB/SIMULINK environment and GSA program is written (in .mfile). The developed model is simulated in a separate program (by .m file using initial population/controller parameters) considering a 10% step load change in area 1. At the first instance, the following parameters are chosen for the application of GSA: population size NP=30; maximum iteration =500; gravitational constants $G_0=30$ and $\alpha =10$; K_0 = total number of agents and decreases linearly to 1 with time [10]. Optimization is terminated by the prespecified number of generations. The flowchart of proposed optimization is shown in Fig. 3.

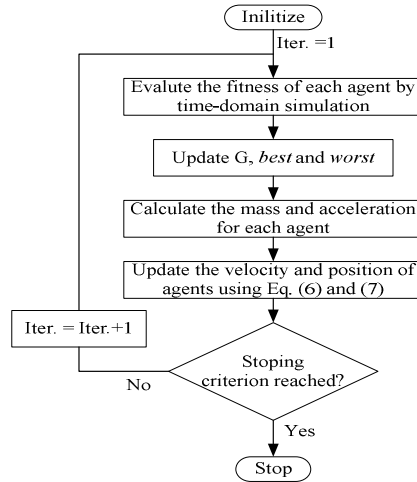


Fig. 3. Flow chart of proposed GSA optimization approach

The optimization was repeated 50 times and the best final solution among the 50 runs is chosen as final controller parameters. The best final solutions obtained in the 50 runs for each objective functions are shown in Table 1. To investigate the effect of objective function on the dynamic performance of the system, settling times (2% of final value) and peak overshoots in frequency and tie-line power deviations along with minimum damping ratios are also provided in Table 1. It can be seen from Table 1 that best system performance is obtained with maximum value of damping ratio and minimum values of settling times and peak overshoots in frequency and tie-line power deviations when ITAE is used as objective function.

Table 1. Tuned controller parameters, settling time, peak overshoot and minimum damping ratio for each objective function

| Objective function | Controller parameters | | $T_s(\text{sec})$ | | | Peak Overshoot | | | ζ |
|--------------------|-----------------------|---------------|-------------------|--------------|------------------|----------------|---------------|------------------|---------------|
| | K_p (-ve) | K_i | ΔF_1 | ΔF_2 | ΔP_{Tie} | ΔF_1 | ΔF_2 | ΔP_{Tie} | |
| ISE | 0.0120 | 0.8641 | 21.30 | 21.30 | 15.10 | 0.0594 | 0.0877 | 0.0152 | 0.0626 |
| IAE | 0.0117 | 0.7668 | 18.00 | 18.00 | 12.90 | 0.0450 | 0.0710 | 0.0119 | 0.0733 |
| ITSE | 0.1228 | 0.7849 | 16.20 | 16.20 | 11.10 | 0.0499 | 0.0838 | 0.0127 | 0.0882 |
| ITAE | 0.1701 | 0.6492 | 12.00 | 11.90 | 8.90 | 0.0390 | 0.0590 | 0.0083 | 0.1144 |

5.2 GSA Parameter Tuning

The success of GSA is heavily dependent on setting of control parameters namely; constant α , initial gravitational constant G_0 , population size NP and number of iteration T. A series of experiments were conducted to properly tune the GSA control parameters in order to optimize the PI parameters employing ITAE objective function. Table 2 shows the GSA outcomes as a result of varying its control parameters.

Table 2. Study of tuning GSA parameters

| Parameter | Min | Ave | Max | St.Dev | Other parameters |
|-------------------------------|------------------|------------------|------------------|------------------|--|
| $\alpha=10$ | 0.6659897 | 0.8980626 | 1.0123918 | 0.1265678 | NP=20,T=50, G ₀ =100 |
| $\alpha=15$ | 0.6659897 | 0.7841139 | 1.0001651 | 0.1267459 | |
| $\alpha=20$ | 0.6659897 | 0.7291873 | 0.9182768 | 0.0802699 | |
| $\alpha=25$ | 0.6749712 | 0.9000713 | 0.9754091 | 0.1228549 | |
| $\alpha=30$ | 0.6870195 | 0.9164638 | 1.0123918 | 0.1375834 | |
| G ₀ =30 | 0.6659945 | 0.8046012 | 1.0123918 | 0.1184529 | NP=20,T=50, $\alpha=20$ |
| G ₀ =70 | 0.6659988 | 0.8512434 | 1.0225763 | 0.1282661 | |
| G₀=100 | 0.6659897 | 0.7291873 | 0.9182768 | 0.0802699 | |
| G ₀ =130 | 0.6680531 | 0.7790509 | 1.0225763 | 0.1055190 | |
| G ₀ =150 | 0.6678854 | 0.7991217 | 1.0035452 | 0.1988937 | |
| NP=10 | 0.6881325 | 0.8231840 | 1.0315944 | 0.1988937 | T=50, $\alpha=20$, G ₀ =100 |
| NP=15 | 0.6706229 | 0.8019416 | 0.9426240 | 0.1250767 | |
| NP=20 | 0.6659897 | 0.7291873 | 0.9182768 | 0.0802699 | |
| NP=25 | 0.6659897 | 0.7280129 | 0.9146560 | 0.0702121 | |
| NP=30 | 0.6659897 | 0.7279858 | 0.9135081 | 0.0679424 | |
| T=30 | 0.6650960 | 0.8781403 | 0.9742163 | 0.1040270 | NP=20,G ₀ =100, $\alpha=20$ |
| T=50 | 0.6659897 | 0.7291873 | 0.9182768 | 0.0802699 | |
| T=100 | 0.6659897 | 0.6774539 | 0.8739539 | 0.0571679 | |
| T=200 | 0.6659897 | 0.6764037 | 0.8739539 | 0.0570019 | |

To quantify the results, 50 independent runs were executed for each parameter variation. It is clear from results shown in Table 2 that the best settings for constant α , gravitational constant G₀, population size NP and number of iteration T are $\alpha=20$, G₀=100, NP=20 and T=100 respectively.

5.3 Analysis of Result

It is clear from Table 3 that with PI structured controller and ITAE objective function (J_4), minimum ITAE value is obtained with GSA (ITAE=0.6659) compared to ITAE values with GA (ITAE=2.74), BFOA (ITAE=1.827) and DE (ITAE=0.9911) techniques. So it can be concluded that for the similar controller structure (PI) and same objective function (ITAE) GSA outperforms GA, BFOA and DE techniques. It is also evident from Table 3 that minimum ITAE value (ITAE=0.1174) is obtained with a PIDF controller and therefore the performance of PIDF controller is superior to that of PI controller. However, when the modified objective function (J_5) given by equation (13) is used better performance is obtained in all respects. The minimum

Table 3. Tuned controller parameter and error with ITAE objective function

| Technique | Tuned controller Parameter | | | | ITAE |
|--------------|----------------------------|----------------|----------------|---------|---------------|
| | K _P | K _I | K _D | N | |
| GA: PI [6] | -0.2346 | 0.2662 | - | - | 2.7474 |
| BFOA: PI [6] | -0.4207 | 0.2795 | - | - | 1.8270 |
| DE: PI [7] | -0.2146 | 0.4335 | - | - | 0.9911 |
| GSA: PI | -0.1880 | 0.6179 | - | - | 0.6659 |
| GSA: PIDF | 1.1884 | 1.9589 | 0.3456 | 54.3260 | 0.1174 |

damping ratio ($\zeta = 0.2374$), ITAE value (ITAE= 1.3096) and settling times (5.17, 6.81 and 4.59 sec for ΔF_1 , ΔF_2 and ΔP_{Tie} respectively) are better compared to those with DE, BFOA and GA technique as shown in Table 4. The best system performance is obtained with GSA optimized PIDF controller optimized using the modified objective function as evident form Table 4. For better visualization of the improvements with the proposed approach, the above results are presented graphically in Fig. 4.

Table 4. Settling time, minimum damping ratio and error with modified objective function

| Technique | Tuned controller Parameter | | | | Settling time T_s (Sec) | | | ζ | ITAE |
|------------------|----------------------------|---------------|---------------|----------------|---------------------------|--------------|------------------|---------------|---------------|
| | K_p | K_i | K_D | N | ΔF_1 | ΔF_2 | ΔP_{tie} | | |
| DE: PI [7] | -0.4233 | 0.2879 | - | - | 5.38 | 6.95 | 6.21 | 0.2361 | 1.6766 |
| GSA: PI | -0.4383 | 0.3349 | - | - | 5.17 | 6.81 | 4.59 | 0.2374 | 1.3096 |
| GSA: PIDF | 1.4011 | 1.9981 | 0.7102 | 93.2760 | 1.92 | 3.19 | 2.86 | 0.4470 | 0.1362 |

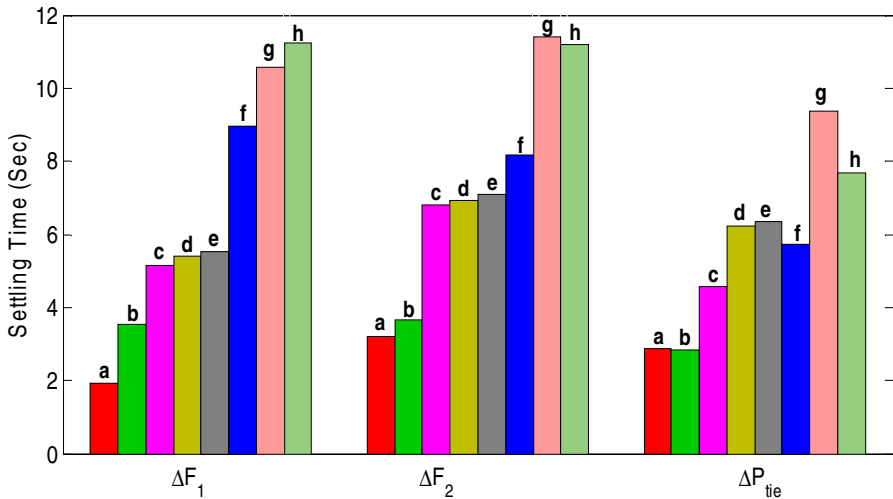


Fig. 4. Comparison of settling time (a) GSA PIDF:J₅ (b) GSA PIDF:J₄ (c) GSA PI:J₅ (d) DE PI:J₅ (e) BFOA PI:J₄ (f) DE PI:J₄ (g) GA PI:J₄ (h) GSA PI:J₄

6 Conclusion

An attempt has been made for the first time to apply a powerful computational intelligence technique like GSA to optimize PI and PIDF controller parameters for AGC of a multi-area interconnected power system. It is observed the performance of the power system is better in terms of minimum damping ratio, settling times and peak overshoots in frequency and tie-line power deviations when ITAE objective function is used compared to IAE, ISTE and ISE objective functions. Then, the parameters of GSA technique are properly tuned and the recommended GSA

parameters are found to be: $\alpha = 20$, $G_0 = 100$, $NP = 20$ and $T = 100$ respectively. Further, a modified objective function is employed and the parameters of PI and PIDF controller are optimized by tuned GSA. The superiority of the proposed method is confirmed by comparing the results with DE, BFOA and GA techniques.

References

- [1] Kundur, P.: Power System Stability and control. TMH, 8th reprint (2009)
- [2] Elgerd, O.I.: Electric energy systems theory. An introduction. Tata McGraw-Hill, New Delhi (1983)
- [3] Ibraheem, Kumar, P., Kothari, D.P.: Recent philosophies of automatic generation control strategies in power systems. *IEEE Trans. Power Syst.* 20(1), 346–357 (2005)
- [4] Shoults, R.R., Jativa Ibarra, J.A.: Multi-area adaptive LFC developed for a comprehensive AGC simulation. *IEEE Trans. Power Syst.* 8(2), 541–547 (1993)
- [5] Khuntia, S.R., Panda, S.: Simulation study for automatic generation control of a multi-area power system by ANFIS approach. *Applied Soft Computing* 12, 333–341 (2012)
- [6] Ali, E.S., Abd-Elazim, S.M.: Bacteria foraging optimization algorithm based load frequency controller for interconnected power system. *Elect. Power and Energy Syst.* 33, 633–638 (2011)
- [7] Rout, U.K., Sahu, R.K., Panda, S.: Design and analysis of differential evolution algorithm based automatic generation control for interconnected power system. *Ain Shams Eng. J.* (2012), <http://dx.doi.org/10.1016/j.asej.2012.10.010>
- [8] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: A Gravitational Search Algorithm. *Information Sciences* 179, 2232–2248 (2009)
- [9] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: Filter modeling using gravitational search algorithm. *Engineering Applications of Artificial Intelligence* 24, 117–122 (2011)
- [10] Li, C., Zhou, J.: Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm. *Energy Conversion and Management* 52, 374–381 (2011)
- [11] Gozde, H., Taplamacioglu, M.C., Kocaarslan, I.: Comparative performance analysis of Artificial Bee Colony algorithm in automatic generation control for interconnected reheat thermal power system. *Int. J. Elect. Power & Energy Syst.* 42, 167–178 (2012)
- [12] Bevrani, H.: Robust Power system frequency control. Springer (2009)

Design and Simulation of FIR High Pass Filter Using Gravitational Search Algorithm

R. Islam¹, Rajib Kar¹, Durbadal Mandal¹, and Sakti Prasad Ghoshal²

¹Department of Electronics and Communication Engineering
National Institute of Technology Durgapur, West Bengal, India- 713209
{rajibkarece, durbadal.bittu}@gmail.com

²Department of Electrical Engineering
National Institute of Technology Durgapur, West Bengal, India- 713209
spghoshalnitdgp@gmail.com

Abstract. In this paper, a linear phase finite impulse response (FIR) high pass (HP) digital filter is designed using a recently proposed heuristic search algorithm called gravitational search algorithm (GSA). Various evolutionary techniques like conventional particle swarm optimization (PSO), differential evolution (DE) and the proposed gravitational search algorithm (GSA) have been applied for the optimal design of linear phase FIR HP filters. Real coded genetic algorithm (RGA) has also been adopted for the sake of comparison. In GSA, agents are considered as objects and their performances are measured by their masses. All these objects attract each other by the gravity forces and these forces cause a global movement of all objects towards the objects with heavier masses. Hence, masses cooperate amongst each other using a direct form of communication through gravitational forces. The heavier masses (which correspond to better solutions) move more slowly than the lighter ones. This guarantees the exploitation step of the algorithm. GSA is apparently free from getting trapped at local optima and premature convergence. Extensive simulation results justify the superiority and optimization efficacy of the GSA over the afore-mentioned optimization techniques for the solution of the multimodal, non-differentiable, highly non-linear, and constrained filter design problems.

1 Introduction

Digital filter is essentially a system or network that improves the quality of a signal and/or extracts information from signals or separates two or more signals which are previously combined. The nature of this filtering action is determined by the frequency response characteristics, which depends on the choice of system parameters, i.e., the coefficients of the difference equations. Thus, by proper selection of the coefficients, one can design frequency selective filters, that pass signals with frequency components in some bands while attenuate signals containing frequency components in other frequency bands. There are different techniques for the design of digital filters, such as window method, frequency sampling method etc. [1]. The major drawback of windowing method is that it does not allow sufficient control on

the frequency response in the various frequency bands and other filter parameters such as transition width and it tends to process relatively long filter lengths. The designer always has to compromise on one or other of the design specifications. Due to several disadvantages of classical optimization methods, evolutionary methods have been adopted in the design of optimal digital filters with better control of parameters and the highest stop band attenuation. Different evolutionary optimization techniques are reported in the literatures like Real Coded GA (RGA) [2-3], orthogonal genetic algorithm (OGA) [4], Tabu search [5], Differential Evolution (DE) [6], Adaptive Differential Evolution (ADE) [7], Particle swarm optimization (PSO) [8-9], some variants of PSO like PSO with Quantum Infusion (PSO-QI) [10], some hybrid algorithms like DE-PSO [11] have been applied for the filter design problem.

Most of the above algorithms show the problems of fixing algorithm's control parameters, premature convergence, stagnation and revisiting of the same solution over and again. In order to overcome these problems, recently, Gravitation search algorithm (GSA) [12] has been applied to IIR filter design problem.

In this paper, the same optimization algorithm GSA and a novel fitness function are employed for the linear phase FIR HP filter design. In the GSA, agents / vectors are considered as objects and their performance is measured by their masses. All these objects attract each other by the gravity forces, and these forces produce a global movement of all objects towards the objects with heavier masses. Hence, masses cooperate using a direct form of communication through gravitational forces. The heavier masses (which correspond to better solutions) move more slowly than lighter ones. This guarantees the exploitation step of the algorithm. GSA does not prematurely restrict the searching space; the searching space varies during the search process. To validate the proposed method, the results obtained by GSA are compared to those of classical optimization techniques like Parks–McClellan (PM) algorithm, and other evolutionary approaches like RGA, conventional PSO, and DE. A comparison of simulation results reveals the optimization efficacy of the GSA over the above optimization techniques for the solution of the multimodal, non-differentiable, highly non-linear, and constrained FIR HP filter design problems.

2 FIR High Pass Filter Design

Digital filters are classified as finite impulse response (FIR) or infinite impulse response (IIR) filters depending upon whether the response of the filter is dependent on only the present inputs or on the present inputs as well as the previous outputs, respectively. A finite-duration impulse response filter has a system function of the form given in (1).

$$H(z) = \sum_{n=0}^N h(n)z^{-n} \quad (1)$$

where $h(n)$ is called impulse response. The FIR filter structures are always stable, and can be designed to have a linear phase response. The impulse responses $h(n)$ are to be determined in the design process and the values of $h(n)$ will determine the type

of the filter, e.g., low pass, high pass etc. The design of filters is based on three broad criteria, namely, the filters should provide as little distortion as possible to the signal; flat pass band; exhibit attenuation characteristics with the highest stop band possible. Other desirable characteristics include narrow filter length, short frequency transition width beyond the cut-off point, and the ability to manipulate the attenuation in the stop band.

In this paper, the GSA and the other algorithms PM, RGA, PSO, DE are individually applied in order to obtain the actual FIR HP filter response as close as possible to the ideal response. The designed filter with $h(n)$ vectors is even symmetric and of even order. The length of $h(n)$ is $N+1$, i.e., the number of coefficients (dimension D) is $N+1$. An ideal filter has a magnitude of 1 in the pass band and a magnitude of 0 in the stop band. The error between the magnitudes of the actual and the ideal frequency spectrums of actual and ideal filters, respectively, is used to evaluate the error fitness value of each vector. At each iteration, the vectors are updated by GSA. Error fitness values of vectors are computed using the new coefficients. The $h(n)$ vector obtained after a certain maximum number of iterations or after the error is below a certain limit is considered to be optimal, resulting in a filter with optimal frequency response. As linear phase FIR filter provides symmetrical coefficients, therefore, the dimension of the problem is halved. The frequency response of the FIR digital filter can be calculated as,

$$H(e^{j\omega_k}) = \sum_{n=0}^N h(n)e^{-j\omega_k n}; \tag{2}$$

where $\omega_k = \frac{2\pi k}{N}$; $H(e^{j\omega_k})$ is the Fourier transform complex vector. This is the FIR filter frequency response. The frequency in $[0, \pi]$ is sampled with N points.

The ideal magnitude response of the filter for the HP filter H_i is given as,

$$H_i(e^{j\omega_k}) = 1 \quad \text{for } \omega_c \leq \omega_k \leq \pi; \\ = 0 \quad \text{otherwise.} \tag{3}$$

In this work, a novel error fitness function has been adopted in order to achieve much higher stop band attenuation. The error fitness function used in this paper is given in (4). Using (4), it is found that the proposed filter design approach results in considerable improvement in frequency responses over the PM, RGA, PSO, DE and other optimization techniques, reported in the prevailing literatures.

$$J = \sum [abs[abs(|H(\omega)|-1)-\delta_p]] + \sum [abs(|H(\omega)|-\delta_s)] \tag{4}$$

where δ_p and δ_s are the ripples in the pass band and the stop band, respectively. For the first term of (4), $\omega \in$ pass band including a portion of the transition band and for the second term of (4), $\omega \in$ stop band including the rest portion of the transition band. The portions of the transition band chosen depend on pass band edge and stop band edge frequencies.

The error fitness function given in (4) represents the generalized fitness function to be minimized using the evolutionary algorithms RGA, conventional PSO, DE and the GSA individually. Each algorithm tries to minimize this error fitness J and thus improves the filter performance. Unlike other error fitness functions [6], [9-11] which consider only the maximum errors, J involves summation of all absolute errors for the whole frequency band, and hence, minimization of J yields much higher stop band attenuation and lesser pass band ripples. Transition width is affected a little. Since the coefficients of the linear phase filter are matched, the dimension of the problem is thus halved. By only determining half of the coefficients, the filter can be designed. This greatly reduces the computational burdens of the algorithms, applied to the optimal design of linear phase FIR filters.

3 Optimization Techniques Employed

3.1 Real Coded Genetic Algorithm (RGA)

Steps of RGA as implemented for optimization of $h(n)$ coefficients are adopted from [2-3]. In this work, initialization of real chromosome string vectors of n_p population, each consisting of a set of $h(n)$ coefficients is made. Size of the set depends on the number of coefficients in a particular filter design.

3.2 Particle Swarm Optimization (PSO)

PSO is a flexible, robust population-based stochastic search or optimization technique with implicit parallelism, which can easily handle with non-differential objective functions, unlike traditional gradient based optimization methods. Mathematically, velocities of the vectors and the vectors are modified according to the equations given in [9].

3.3 Differential Evolution (DE) Algorithm

DE was proposed by Rainer Storn and Price in 1997 [16]. DE is a population based, biological evolutionary strategy intimated, stochastic optimization algorithm that searches the solution space by using the weighted difference between two population vectors to determine the third vector. Apart from randomly generated population vectors, DE goes through Mutation, Crossover and Selection in each iteration cycle and then repeats again till the maximum iteration cycles.

3.4 Gravitational Search Algorithm

The limitations of PSO and DE are that they may be influenced by parameter convergence and stagnation problem. To overcome the problems associated with DE

and PSO; this paper adopts one recently proposed novel heuristic optimization algorithm known as gravitational search algorithm (GSA) for the purpose of FIR digital filter design.

In GSA [12-15], agents / solution vectors are considered as objects and their performance is measured by their masses. All these objects attract each other by the gravity forces, and these forces cause a global movement of all objects towards the objects with heavier masses. Hence, masses cooperate using a direct form of communication through gravitational forces. The heavier masses (which correspond to better solutions) move more slowly than lighter ones. This guarantees the exploitation step of the algorithm.

One way to perform a good compromise between exploration and exploitation is to reduce the number of agents with lapse of time. Hence, it is supposed that a set of agents with bigger masses apply their forces to the other. However, this policy is to be adopted carefully because it may reduce the exploration power and increase the exploitation capability. The details are given in [12-13].

The steps of the algorithm are depicted in Table 1.

Table 1. Steps of GSA as implemented for linear phase FIR HP filter design

| | |
|---------|--|
| step1. | Initialization: Population (swarm size) of agent vectors, $n_p=25$; maximum iteration cycles = 200; number of filter coefficients $h(n) = (nvar/2 + 1)$, filter order = nvar (even and symmetric for linear phase FIR filter); minimum and maximum values of filter coefficients, $h_{min} = -2$, $h_{max} = 2$; number of samples=128; $\delta_p = 0.1$, $\delta_s = 0.01$; $\alpha = 20$; $G_0 = 1000$; $rNorm=2$; $rPower=1$; $velo=zeros(n_p, nvar/2+1)$; $\epsilon = 0.0001$. |
| step2. | Generate initial agent vectors $h(n)$ of filter coefficients ($nvar/2 + 1$) randomly within limits. |
| step3. | Computation of error fitness values of the total population, n_p , as defined by (4). |
| step4. | Computation of the population based best solution (h_{gbest}) vector. |
| step5. | Update $G(t)$, $best(t)$, $worst(t)$ and $M_i(t)$ for $i = 1, 2, \dots, n_p$; t is current iteration cycle. |
| step6. | Calculation of the total forces in different directions. |
| step7. | Calculation of accelerations and velocities of agents. |
| step8. | Updating agents' positions. |
| step9. | Repeat Steps 3 to 8 until the stopping criterion (either maximum iteration cycles or near global optimal solution or agent, h_{gbest}) is met. |
| step10. | Finally, h_{gbest} is the vector of optimal filter coefficients ($nvar/2 + 1$). Form complete nvar coefficients by copying (because the filter is even symmetric and has linear phase characteristic) before getting the final FIR HP optimal frequency spectrum. |

4 Results and Discussions

In order to demonstrate the effectiveness of the proposed optimal filter design method, optimal designs of FIR HP filters of two different orders 20 and 24 are realized using PM, RGA, PSO, DE and GSA individually. The MATLAB simulation has been performed extensively to realize the FIR HP filters. The numbers of the respective filter coefficients are 21 and 25. The sampling frequency has been chosen as $f_s = 1\text{Hz}$. Also, for all the simulations the number of sampling points is taken as 128. Each algorithm is run for 100 times to get the best solutions. The best results are reported in this work. Table 2 shows the best chosen parameters used for different optimization algorithms. The parameters of the HP filters to be designed are: pass band ripple (δ_p) = 0.1; stop band ripple (δ_s) = 0.01; pass band (normalized) edge frequency (ω_p) = 0.55; stop band (normalized) edge frequency (ω_s) = 0.45; and transition width=0.1. The best optimized coefficients for the designed HP filter of orders 20 and 24 have been individually computed by PM algorithm, RGA, PSO, DE and GSA.

Table 2. RGA, PSO, DE, GSA parameters

| Parameters | RGA | PSO | DE | GSA |
|-----------------------------------|------------------------|-----------|----------|-------------------|
| Population size | 120 | 25 | 25 | 25 |
| Iteration cycles | 100 | 100 | 100 | 100 |
| Crossover rate | 1.0 | - | - | |
| Crossover | Single Point Crossover | - | - | |
| Mutation rate | 0.01 | - | - | |
| Mutation | Gaussian Mutation | - | - | |
| Selection, probability | Roulette, 1/3 | - | - | |
| $C_1 = C_2$ | - | 2.05 | | |
| v_i^{\min}, v_i^{\max} | - | 0.01, 1.0 | | |
| w_{\max}, w_{\min} | - | 1.0, 0.4 | | |
| C_r, F | - | - | 0.3, 0.5 | |
| $\alpha, G_0,$ $rNORM, rPower$ | | | | 20, 1000, 2, 1 |
| ϵ | | | | 0.0001 |

Table 3. Comparative results of performance parameters of all algorithms for FIR HP filter of order 20

| Algorithm | FIR HP filter of order 20 | | | | |
|-----------|------------------------------------|--|--|-------------------------------|-----------------------------------|
| | Maximum stop band attenuation (dB) | Maximum, average pass band ripple (normalized) | Maximum, average stop band ripple (normalized) | Transition width (normalized) | Execution Time for 100 cycles (s) |
| PM | 23.55 | 0.066, 0.066 | 0.06642, 0.06632 | 0.0837 | - |
| RGA | 25.25 | 0.119, 0.1137 | 0.05461, 0.0248 | 0.0863 | 4.5116 |
| PSO | 28.1 | 0.124, 0.119 | 0.03935, 0.01919 | 0.0863 | 3.0249 |
| DE | 29.16 | 0.137, 0.127 | 0.03483, 0.02605 | 0.0875 | 3.7754 |
| GSA | 31.78 | 0.143, 0.125 | 0.02577, 0.01599 | 0.0896 | 2.1011 |

Table 4. Statistical parameters of stop band attenuation for different algorithms

| Algorithm | FIR HP filter of order 20 | | | | FIR HP filter of order 24 | | | |
|-----------|----------------------------|--------|----------|--------------------|----------------------------|--------|----------|--------------------|
| | Stop Band Attenuation (dB) | | | | Stop Band Attenuation (dB) | | | |
| | Maximum | Mean | Variance | Standard Deviation | Maximum | Mean | Variance | Standard Deviation |
| PM | 23.55 | 23.564 | 0.00010 | 0.0101 | 23.55 | 23.556 | 0.000024 | 0.00489 |
| RGA | 25.25 | 32.095 | 20.975 | 4.579 | 27.85 | 35 | 12.472 | 3.531 |
| PSO | 28.1 | 35.39 | 16.1874 | 4.0234 | 28.65 | 35.092 | 11.5287 | 3.395 |
| DE | 29.16 | 37.056 | 18.312 | 4.279 | 35.39 | 38.22 | 3.1978 | 1.788 |
| GSA | 31.78 | 36.292 | 5.864 | 2.4216 | 32.84 | 36.145 | 4.1837 | 2.0454 |

Table 5. Comparative results of performance parameters of all algorithms for FIR HP filter of order 24

| Algorithm | FIR HP filter of order 24 | | | | |
|-----------|------------------------------------|--|--|-------------------------------|-----------------------------------|
| | Maximum stop Band attenuation (dB) | Maximum, average pass band ripple (normalized) | Maximum, average stop band ripple (normalized) | Transition width (normalized) | Execution Time for 100 cycles (s) |
| PM | 23.55 | 0.066, 0.066 | 0.06642, 0.06637 | 0.0835 | - |
| RGA | 27.85 | 0.131, 0.118 | 0.0405, 0.01807 | 0.0725 | 5.1309 |
| PSO | 28.65 | 0.12, 0.1143 | 0.0369, 0.01744 | 0.0743 | 3.7564 |
| DE | 30.23 | 0.119, 0.10033 | 0.0313, 0.01566 | 0.0813 | 4.2963 |
| GSA | 32.84 | 0.121, 0.115 | 0.0228, 0.01518 | 0.0778 | 2.9133 |

Figure 1 depicts the magnitude (dB) plots for the FIR HP filter of order 20 designed individually by the above algorithms. Table 3 summarizes the comparative results of different performance parameters like, maximum stop band attenuation, maximum and average pass band ripples, maximum and average stop band ripples and transition width of the HP filter of order 20, designed using PM, RGA, PSO, DE and GSA individually. Table 4 shows the statistical data like mean, variance and the standard deviation of the stop band attenuation for the designed HP filters of orders 20 and 24 using the above algorithms.

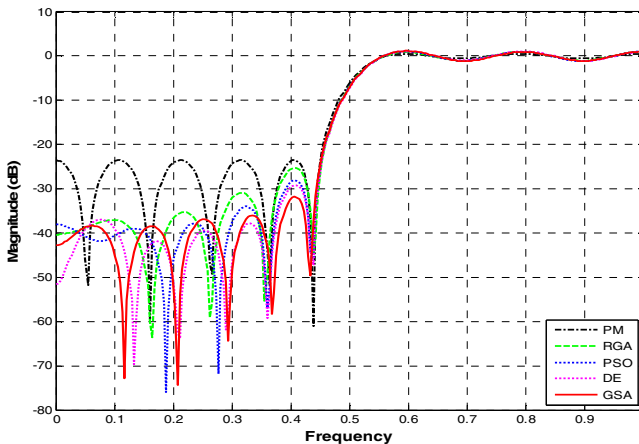


Fig. 1. Magnitude (dB) plot for the FIR HP filter of order 20

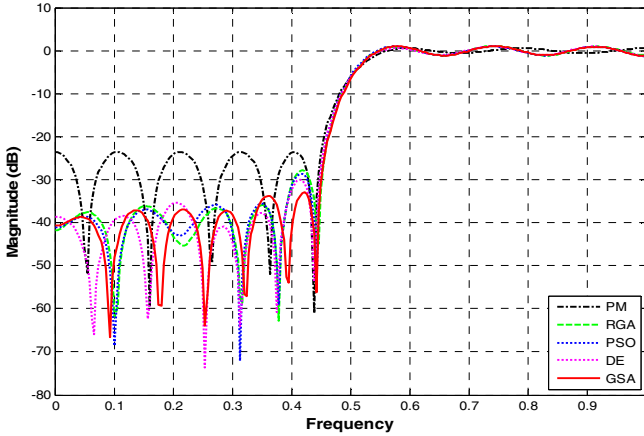


Fig. 2. Magnitude Plot (dB) for FIR HP Filter of order 24

Figure 2 depicts the magnitude (dB) plots for the FIR HP filter of order 24 designed individually by the above algorithms. Table 5 summarizes the comparative results of different performance parameters like, maximum stop band attenuation, maximum and average pass band ripples, maximum and average stop band ripples and transition width of the HP filter of order 24, designed using PM, RGA, PSO, DE and GSA. Table 6 shows the comparative best results of GSA and other reported works in regard to the FIR HP filter design.

Table 3 shows that the proposed GSA based approach for the 20th order FIR HP filter design results in the highest 31.78 dB stop band attenuation, maximum pass band ripple (normalized) of 0.143, average pass band ripple (normalized) of 0.125, the lowest maximum stop band ripple (normalized) of 0.02577, average stop band ripple (normalized) = 0.01599 and transition width = 0.0896.

Table 4 shows that the variance and the standard deviation of stop band attenuation for the FIR HP filters of orders 20 and 24, respectively, designed using GSA are the lowest among other aforementioned evolutionary optimization algorithms. Table 5 shows that the proposed GSA based approach for the 24th order FIR HP filter design results in the highest 32.84 dB stop band attenuation, maximum pass band ripple (normalized) of 0.121, the lowest maximum stop band ripple (normalized) of 0.0228, average pass band ripple (normalized) = 0.115, average stop band ripple (normalized) = 0.01518 and transition width of 0.0778. From the figures and tables, it is evident that with almost same level of the transition width, the proposed GSA based filter design approach produces the highest stop band attenuation (dB) and the lowest stop band ripple. So, in the stop band region, the filters designed by GSA results in the best responses as compared to PM, RGA, PSO and DE. From Table 6, one can infer that the GSA based filter design approach is the best so far among those of the literatures available for this purpose.

Table 6. Comparison of GSA’s results with other reported results

| Model | Parameter | | | | | |
|-------|-------------|-------|------------------------------------|---------------------------------------|---------------------------------------|------------------|
| | Filter type | Order | Maximum stop band attenuation (dB) | Maximum pass band ripple (normalized) | Maximum stop band ripple (normalized) | Transition width |
| [6] | Low pass | 20 | NR* | >0.08 | >0.09 | >0.16 |
| [11] | Low Pass | 20 | <23 dB | 0.291 | 0.270 | >0.13 |
| [8] | Low Pass | 33 | <29dB | NR* | NR* | NR* |
| | Band Pass | 33 | <25 dB | NR* | NR* | NR* |
| [9] | Low pass | 30 | <30dB(Approx.) | 0.15 | 0.031 | 0.05 |
| [10] | Low Pass | 20 | < 27dB | >0.1 | >0.06 | >0.15 |
| | Band pass | 20 | <8dB | >0.2 | >0.05 | >0.07 |
| GSA | High Pass | 20 | 31.78 | 0.143 | 0.02577 | 0.0907 |
| | High Pass | 24 | 32.84 | 0.121 | 0.0228 | 0.0778 |

NR* means not reported in the referred literature

4.1 Comparative Effectiveness and Convergence Profiles of RGA, PSO, DE and GSA

In order to compare the algorithms in terms of the error fitness value, Figure 3 shows the convergences of error fitness values obtained by RGA, PSO, DE and GSA. The convergence profiles are shown for the FIR HP filter of order 24. As shown in Figure 3, RGA converges to the minimum error fitness value of 2.70 in 5.1309s; PSO converges to the minimum error fitness value of 2.0830 in 3.7564s; DE converges to the minimum error fitness value of 1.5350 in 4.2963s; whereas the GSA converges to the minimum error fitness value of 0.9686 in the lowest time 2.9133s. The above mentioned execution times for 100 cycles may be verified from Table 5.

The similar convergence profiles have been achieved for the FIR HP filter of order 20, which are not shown here. The execution times for 100 cycles for RGA, PSO, DE and GSA, respectively, are shown in Table 3 for the FIR HP filter of order 20.

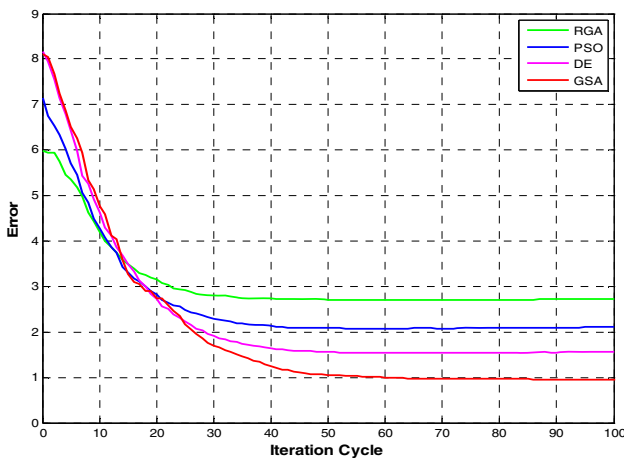


Fig. 3. Convergence Profiles for RGA, PSO, DE and GSA in case of 24th Order FIR HP Filter

Hence, for the FIR HP filter designs, GSA converges to the least minimum error fitness values in the lowest execution times for finding the optimum filter coefficients. GSA's lowest execution time is because of its very few easy computational steps devoid of any mutation, crossover, selection etc., which are present in other algorithms. With a view to the above fact, it may finally be inferred that the performance of the GSA is the best among all algorithms. All optimization programs are run in MATLAB 7.5 version on core (TM) 2 duo processor, 3.00 GHz with 2 GB RAM.

5 Conclusions

In this paper, a meta-heuristic gravitational search algorithm (GSA) is applied to the solution of the constrained, multi-modal, non-differentiable, and highly nonlinear FIR high pass filter design problem, which achieves optimal filter coefficients and the highest stop band attenuation and the lowest stop band ripples with a little increase or change in the transition width. It is revealed that GSA converges very fast to the best quality near optimal solution and possesses the best convergence characteristics, i.e., the lowest minimum error fitness value among other prevailing algorithms reported in this work. It is also evident from the results obtained by a large number of trials that GSA is consistently free from the shortcoming of premature convergence exhibited by the other optimization algorithms. The simulation results clearly reveal that GSA may be used as a good optimizer for the solution of obtaining the optimal filter coefficients in a practical digital filter design problem in digital signal processing systems.

References

1. Oppenheim, A.V., Schafer, R.W., Buck: *Discrete Time Signal Processing*, 2nd edn. Prentice Hall (1999)
2. Ahmad, S.U., Antoniou, A.: A genetic algorithm approach for fractional delay FIR filters. In: *IEEE Int. Symp. on Circuits and Systems, ISCAS 2006*, pp. 2517–2520 (2006)
3. Lu, H.C., Tzeng, S.-T.: Design of arbitrary FIR log filters by genetic algorithm approach. *Signal Processing* 80, 497–505 (2000)
4. Ahmad, S.U., Andreas, A.: Cascade-Form Multiplier less FIR Filter Design Using Orthogonal Genetic Algorithm. In: *IEEE Int. Symp. on Signal Proc. and Info. Tech.*, pp. 932–937 (2006)
5. Karaboga, D., Horrocks, D.H., Karaboga, N., Kalinli, A.: Designing digital FIR filters using Tabu search algorithm. In: *IEEE Int. Symp. on Circuits and Systems, ISCAS 1997*, vol. 4, pp. 2236–2239 (1997)
6. Karaboga, N., Cetinkaya, B.: Design of digital FIR filters using differential Evolution algorithm. *Circuits Systems Signal Processing* 25(5), 649–660 (2006)
7. Pan, S.-T.: Evolutionary Computation on Programmable Robust IIR Filter Pole-Placement Design. *IEEE Trans. on Inst. and Meas.* 60(4), 1469–1479 (2011)
8. Najjarzadeh, M., Ayatollahi, A.: FIR Digital Filters Design: Particle Swarm Optimization Utilizing LMS and Minimax Strategies. In: *IEEE Int. Symp. on Signal Proc. and Info. Tech.*, ISSPIT 2008, pp. 129–132 (2008)

9. Ababneh, J.I., Bataineh, M.H.: Linear phase FIR filter design using particle swarm optimization and genetic algorithms. *Digital Signal Processing* 18(4), 657–668 (2008)
10. Sarangi, A., Mahapatra, R.K., Panigrahi, S.P.: DEPSO and PSO-QI in digital filter design. *Expert Sys. with Applications* 38(9), 10966–10973 (2011)
11. Luitel, B., Venayagamoorthy, G.K.: Differential evolution particle swarm optimization for digital filter design. In: *IEEE Congress on Evolutionary Computation, CEC 2008*, pp. 3954–3961 (2008)
12. Rashedi, E., Hossien, N., Saryazdi, S.: Filter modelling using gravitational search algorithm. *Engineering Appl. of Artificial Intelligence* 24(1), 117–122 (2011)
13. Chatterjee, A., Ghoshal, S.P., Mukherjee, V.: A maiden application of gravitational search algorithm with wavelet mutation for the solution of economic load dispatch problems. *International Journal of Bio-inspired Computation* 4(1), 33–46 (2012)
14. Saha, S.K., Ghoshal, S.P., Kar, R., Mandal, D.: Design and Simulation of FIR Band Pass and Band Stop Filters using Gravitational Search Algorithm. *Journal of Memetic Computing* (2013), doi:10.1007/s12293-013-0122-6
15. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: A Gravitational Search Algorithm. *Information Sciences* 179, 2232–2248 (2009)
16. Storn, R., Price, K.: *Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces*. Technical report, International Computer Science Institute, Berkley, TR-95-012 (1995)

Solution of Optimal Reactive Power Dispatch by an Opposition-Based Gravitational Search Algorithm

Binod Shaw¹, V. Mukherjee², and Sakti Prasad Ghoshal³

¹ Department of Electrical Engineering, Asansol Engineering College, Asansol,
West Bengal, India

² Department of Electrical Engineering, Indian School of Mines, Dhanbad, Jharkhand, India

³ Department of Electrical Engineering, National Institute of Technology, Durgapur,
West Bengal, India

{binodshaw2000, spghoshalnitdgp}@gmail.com,
vivek_agamani@yahoo.com

Abstract. Management of reactive power resources is vital for stable and secure operation of power systems in the view point of voltage stability. In the present work, opposition-based gravitational search algorithm (OGSA) is applied for the solution of optimal reactive power dispatch (ORPD) of power systems. ORPD is an optimisation problem that decreases grid congestion with one or more objective of minimising the active power loss for a fixed economic power schedule. In this study, OGSA is tested on the standard IEEE 30-bus test system with different test cases such as minimisation of active power losses, improvement of voltage profile and enhancement of voltage stability. The obtained results are compared with those reported in the literature. Simulation results demonstrate the superiority and accuracy of the proposed algorithm. Considering the quality of the solution obtained, the proposed algorithm seems to be effective and robust to solve the ORPD problem.

Keywords: Opposition-based gravitational search algorithm, optimal reactive power dispatch.

1 Introduction

In power systems, basic objective of optimal reactive power dispatch (ORPD) is to identify optimal setting of control variables which minimize the given objective function as either total transmission line loss (P_{Loss}), or absolute value of total voltage deviation (TVD), or improvement of voltage stability index (VSI) while satisfying the unit and system constraints. This goal is accomplished by proper adjustment of reactive power variables like generator voltage magnitudes, transformer tap settings and switchable VAR sources [1]. Increased interests are being paid by the researchers towards the application of artificial intelligence technology for the solution of the ORPF problems. Genetic algorithm (GA) [2], improved GA [3], adaptive GA [4], evolutionary programming (EP) [5], particle swarm optimization (PSO) [6], hybrid PSO [7], bacterial foraging optimization (BFO) [8], differential

evolution (DE) [9-11], seeker optimization algorithm (SOA) [12], gravitational search algorithm (GSA) [13] are just a few among the numerous techniques reported in the recent literatures. Blending of wavelet mutation with GSA is presented in [14]. A novel dynamic constrained optimization with offspring repair based GSA is introduced in [15]. Which have been used for the solution of the ORPD problems of power system.

In the present work, a new opposition-based GSA (OGSA) [16] is applied for the solution of the ORPD problems of power system. The performance of the proposed hybrid algorithm is tested on standard IEEE 30-bus test system for the ORPD problems. The potential and effectiveness of the proposed approach are demonstrated by comparing the results to those reported in the recent literatures.

The rest of the paper is organized as follows. In Section 2, mathematical problem formulation is done. Proposed optimization algorithm and its application to the ORPD problems are narrated in Section 3. Numerical examples and simulation results are presented in Section 4 to demonstrate the performance of the proposed algorithm for the ORPD problems. Section 5 focuses on the conclusion of the present work.

2 Mathematical Problem formulation

2.1 Minimization of Real Power Loss

The objective of the reactive power optimization is to minimize the active power loss in the transmission network, which can be defined as follows [12]

$$\text{Min } P_{Loss} = f(\vec{X}_1, \vec{X}_2) = \sum_{k \in N_E} g_k (V_i^2 + V_j^2 - 2V_i V_j \cos \theta_{ij}) \quad (1)$$

where $f(\vec{X}_1, \vec{X}_2)$ denotes the active power loss function of the transmission network; \vec{X}_1 is the control variable vector $[V_G \ T_k \ Q_C]^T$; \vec{X}_2 is the dependent variable vector; V_G is the generator voltage vector (continuous) except the slack bus voltage; T_k is the transformer tap vector (integer); Q_C is the shunt capacitor/inductor vector (integer); V_L is the load bus voltage vector; Q_G is the generator reactive power vector; $k = (i, j)$, $i \in N_B, j \in N_i$, g_k is the conductance of branch k ; θ_{ij} is the voltage angle difference between buses i and j ; P_{G_i} is the injected active power at bus i ; P_{D_i} is the demanded active power at bus i ; V_i is the voltage at bus i ; G_{ij} is the transfer conductance between bus i and j ; B_{ij} is the transfer susceptance between buses i and j ; Q_{G_i} is the injected reactive power at bus i ; Q_{D_i} is the demanded reactive power at bus i ; N_E is the of number of network branches; N_{PQ} is the of number of PQ buses; N_B is the of number of total buses; N_i is the set of numbers of buses adjacent to bus i (including bus i); N_0 is the of number of total buses excluding

slack bus; N_C is the of number of possible reactive power source installation buses; N_G is the of number of generator buses; N_T is the of number of transformer branches; S_l is the power flow in branch l ; the superscripts “min” and “max” in (1) denote the corresponding lower and upper limits, respectively.

2.2 Minimization of TVD

Minimization of TVD of load buses can allow the improvement of voltage profile [1]. This objective function may be formulated as follows

$$\text{Min TVD} = \sum_{i \in N_L} |V_i - V_i^{ref}| \quad (2)$$

where N_L is the number of load buses in the power system and V_i^{ref} is the desired value of the voltage magnitude of the i th bus which is equal to 1.0 p.u.

2.3 Improvement of VSI

The VSI objective may mathematically be expressed as:

$$L = \min(L_{\max}) = \min[\max(L_k)] \quad \text{where } k = 1, 2, \dots, N_L \quad (3)$$

$$L_k = \left| 1 - \sum_{i \in N_G} F_{ji} \frac{V_i}{V_j} \angle \{\alpha_{ij} + (\delta_i - \delta_j)\} \right|$$

where L_k is the voltage stability indicator (L -index) of the k th node and may be formulated as [9]:

$$L_k = \left| 1 - \sum_{i \in N_G} F_{ji} \frac{V_i}{V_j} \angle \{\alpha_{ij} + (\delta_i - \delta_j)\} \right| \quad (4)$$

where F_{ji} is the (i, j) th components of the sub matrix obtained by the partial inversion of Y_{BUS} and is given by

$$F_{ji} = -[Y_{jj}]^{-1}[Y_{ji}] \quad (5)$$

where Y_{jj} is the self-admittance of the j th bus; Y_{ji} is the mutual admittance between the j th bus and the i th bus; α_{ij} is the phase angle of the term F_{ij} ; δ_i , δ_j are the phase angle of the i th and the j th bus voltages, respectively.

2.4 Equality and Inequality Constraints of ORPD Problems

It is worth mentioning that during the process of optimization, all the constraints are satisfied as explained below [1].

- The load flow equality constraints are satisfied by power flow algorithm.
- The generator bus voltage (V_{Gi}), transformer tap setting (T_k) and switchable reactive power compensations (Q_{Ci}) are optimization variables and these

are self-restricted between their respective minimum and maximum values by the algorithm.

- (c) The limits on active power generation at the slack bus ($P_{G slack}$), load bus voltages (V_{Li}), reactive power generation (Q_{Gi}) and transmission line flow (S_{li}) are state variables. These are restricted by adding a penalty function to the objective functions.

3 Proposed Optimization Algorithm and Its Application to ORPD Problem

3.1 Gravitational Search Algorithm

Rashedi *et al.* proposes gravitational search algorithm (GSA) in [17]. Based on GSA, mass of each agent is calculated after computing current population's fitness given in (6)-(7)

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (6)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{i=1}^{Np} m_i(t)} \quad (7)$$

where $worst(t)$ and $best(t)$ are defined in (8)-(9).

$$best(t) = \min_{j \in \{1, \dots, n\}} fit_j(t) \quad (8)$$

$$worst(t) = \max_{j \in \{1, \dots, n\}} fit_j(t) \quad (9)$$

Total forces applied on an agent from a set of heavier masses should be considered based on the law of gravity as stated in (10) which is followed by calculation of acceleration using the law of motion as presented in (11). Afterwards, next velocity of an agent, (as given in (12)), is calculated as a fraction of its current velocity added to its acceleration. Then, its next position may be calculated by using (13).

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i}^{Np} rand_j \times G(t) \times \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \epsilon} \times (x_j^d(t) - x_i^d(t)) \quad (10)$$

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} = \sum_{j \in Kbest, j \neq i}^{Np} rand_j \times G(t) \times \frac{M_j(t)}{R_{ij}(t) + \epsilon} \times (x_j^d(t) - x_i^d(t)) \quad (11)$$

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \quad (12)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (13)$$

In GSA, the gravitational constant (G) will take an initial value (G_0), and it will be reduced with time as given in (14)

$$G(t) = G_0 \times e^{-\tau \left(\frac{iter}{iter \max} \right)} \quad (14)$$

3.2 Opposition-Based GSA

Tizhoosh introduced the concept of *opposition-based learning* (OBL) in [18]. The steps of the proposed OGSA algorithm are enumerated in Fig. 1

- Step 1** Read the parameters of power system and those of OGSA and specify the lower and upper limits of each variable.
- Step 2** Population-based initialization (P_0).
- Step 3** Opposition-based population initialization (OP_0).
- Step 4** Select N_p fittest individuals from set of $\{P_0, OP_0\}$ as initial population P_0 .
- Step 5** Fitness evaluation of the agents using the objective function of the problem based on the results of Newton–Raphson power flow analysis [19].
- Step 6** Update $M_i(t)$ based on (6)-(7), $best(t)$ based on (8), and $worst(t)$ based on (9), and $G(t)$ based on (14) for $i = 1, 2, \dots, N_p$.
- Step 7** Calculation of the total forces in different directions by using (10).
- Step 8** Calculation of acceleration by (11) and the velocity by (12).
- Step 9** Updating agents' positions by (13).
- Step 10** Check for the constraints of the problem.
- Step 11** Opposition based generation jumping.
- Step 12** Go to Step 5 until a stopping criterion is satisfied.

Fig. 1. Pseudo code of the proposed OGSA algorithm

4 Numerical Examples and Solution Results

The proposed hybrid OGSA has been applied to solve the ORPD problems of IEEE 30-bus test power system. The software has been written in MATLAB 2008a language and executed on a 2.63 GHz Pentium IV personal computer with 3 GB RAM.

4.1 Test System

IEEE 30-bus power system is taken as test system. The system input data are given in [13]. This test system has nineteen control variables and these are six generator

voltage magnitudes (at the buses 1, 2, 5, 8, 11 and 13), four transformers with off-nominal tap ratio (at lines 6–9, 6–10, 4–12 and 28–27) and nine shunt VAR compensation devices (at buses 10, 12, 15, 17, 20, 21, 23, 24 and 29). The total system demand is 2.834 p.u. at 100 MVA base

4.1.1 Minimization of System P_{Loss} for IEEE 30-Bus Power System

The proposed approach is applied for minimization of P_{Loss} as one of the objective function. The obtained optimal values of control variables, as obtained from the proposed OGSA method are given in Table I. The results obtained by the proposed OGSA are compared to those reported in the literature like GSA [13], biogeography-based optimization (BBO) [20], DE [9], comprehensive learning PSO (CLPSO) [21], PSO [21] and self-adaptive real coded GA (SARGA) [22]. The obtained minimum P_{Loss} from the proposed approach is 4.4984 MW. The value of P_{Loss} yielded by OGSA is less by 0.01591 MW (i.e. 0.3524%) compared to GSA-based best results of 4.514310 MW. Comparative DE-, GSA- and OGSA-based convergence profile of P_{Loss} (MW) for this test power system is presented in Fig.2. From this figure it may be observed that the convergence this test system is promising one. The computational times of the compared algorithms are also shown in Table I. It may be seen from this table that the computing time of OGSA is less than other evolutionary meta-heuristics including the basic GSA reported in [13].

Table 1. Comparison of simulation results for IEEE 30-bus test power system with P_{Loss} minimization objective

| Variable | OGSA | Variable | OGSA |
|------------------------------|--------|------------------------|---------------|
| Generator voltage | | Capacitor banks | |
| v_1 , p.u. | 1.0500 | Q_{C-10} , p.u. | 0.0330 |
| v_2 , p.u. | 1.0410 | Q_{C-12} , p.u. | 0.0249 |
| v_5 , p.u. | 1.0154 | Q_{C-15} , p.u. | 0.0177 |
| v_8 , p.u. | 1.0267 | Q_{C-17} , p.u. | 0.0500 |
| v_{11} , p.u. | 1.0082 | Q_{C-20} , p.u. | 0.0334 |
| v_{13} , p.u. | 1.0500 | Q_{C-21} , p.u. | 0.0403 |
| Transformer tap ratio | | Q_{C-23} , p.u. | 0.0269 |
| T_{6-9} | 1.0585 | Q_{C-24} , p.u. | 0.0500 |
| T_{6-10} | 0.9089 | Q_{C-29} , p.u. | 0.0194 |
| T_{4-12} | 1.0141 | P_{Loss} , MW | 4.4984 |
| T_{28-27} | 1.0182 | TVD, p.u. | 0.8085 |
| CPU time, s | 89.19 | L-index, p.u. | 0.1407 |

Table 2. Statistical comparison of results of IEEE 30-bus power test system based with P_{Loss} minimization

| Variable | OGSA | GSA [13] | BBO [20] | DE [9] | CLPSO [21] | PSO [21] | SARGA [22] |
|--------------------|---------------|-------------|-------------|-----------|---------------|-------------|---------------|
| P_{Loss} , (MW) | 4.4984 | 4.514310 | 4.5511 | 4.5550 | 4.5615 | 4.6282 | 4.57401 |
| TVD, (p.u.) | 0.8085 | 0.875220 | NR* | 1.9589 | 0.4773 | 1.0883 | NR* |
| L -index, (p.u.) | 0.1407 | 0.141090 | NR* | 0.5513 | NR* | NR* | NR* |
| CPU time, (s) | 89.19 | 94.6938 | NR* | NR* | 138 | 130 | NR* |

NR* means not reported

Table 3. Comparison of simulation results for IEEE 30-bus test power system with (a) TVD minimization objective and (b) improvement of VSI

| Variable | (a) TVD minimization | | | (b) Improvement of VSI | | |
|-----------------------|----------------------|----------|--------|------------------------|----------|--------|
| | OGSA | GSA [13] | DE [9] | OGSA | GSA [13] | DE [9] |
| Generator voltage | | | | | | |
| V_1 , p.u. | 0.9746 | 0.983850 | 1.0100 | 1.0951 | 1.100000 | 1.0993 |
| V_2 , p.u. | 1.0273 | 1.044807 | 0.9918 | 1.0994 | 1.100000 | 1.0967 |
| V_5 , p.u. | 0.9965 | 1.020353 | 1.0179 | 1.0991 | 1.100000 | 1.0990 |
| V_8 , p.u. | 0.9982 | 0.999126 | 1.0183 | 1.0991 | 1.100000 | 1.0346 |
| V_{11} , p.u. | 0.9826 | 1.077000 | 1.0114 | 1.0995 | 1.100000 | 1.0993 |
| V_{13} , p.u. | 1.0403 | 1.043932 | 1.0282 | 1.0994 | 1.100000 | 0.9517 |
| Transformer tap ratio | | | | | | |
| T_{6-9} | 0.9909 | 0.900000 | 1.0265 | 0.9728 | 0.900000 | 0.9038 |
| T_{6-10} | 1.0629 | 1.100000 | 0.9038 | 0.9000 | 0.900000 | 0.9029 |
| T_{4-12} | 1.0762 | 1.050599 | 1.0114 | 0.9534 | 0.900000 | 0.9002 |
| T_{28-27} | 1.0117 | 0.961999 | 0.9635 | 0.9501 | 1.019538 | 0.9360 |
| Capacitor banks | | | | | | |
| Q_{C-10} , p.u. | 0.0246 | 0.000000 | 4.9420 | 0.0021 | 5.000000 | 0.6854 |
| Q_{C-12} , p.u. | 0.0175 | 0.473512 | 1.0885 | 0.0265 | 5.000000 | 4.7163 |
| Q_{C-15} , p.u. | 0.0283 | 5.000000 | 4.9985 | 0.0000 | 5.000000 | 4.4931 |
| Q_{C-17} , p.u. | 0.0403 | 0.000000 | 0.2393 | 0.0006 | 5.000000 | 4.5100 |
| Q_{C-20} , p.u. | 0.0000 | 5.000000 | 4.9958 | 0.0000 | 5.000000 | 4.4766 |
| Q_{C-21} , p.u. | 0.0270 | 0.000000 | 4.9075 | 0.0000 | 5.000000 | 4.6075 |
| Q_{C-23} , p.u. | 0.0385 | 4.999834 | 4.9863 | 0.0000 | 5.000000 | 3.8806 |
| Q_{C-24} , p.u. | 0.0257 | 5.000000 | 4.9663 | 0.0009 | 5.000000 | 3.8806 |
| Q_{C-29} , p.u. | 0.0000 | 5.000000 | 2.2325 | 0.000 | 5.000000 | 3.2541 |
| P_{Loss} , MW | 6.9044 | 6.911765 | 6.4755 | 5.9198 | 4.975298 | 7.0733 |
| TVD, p.u. | 0.0640 | 0.067633 | 0.0911 | 1.9887 | 0.215793 | 1.419 |
| L -index, p.u. | 0.13381 | 0.134937 | 0.5734 | 0.1230 | 0.136844 | 0.1246 |
| CPU time, s | 190.14 | 198.6532 | NR* | 185.16 | NR* | NR* |

NR* means not reported

4.12. Minimization of System TVD for IEEE 30-Bus Power System

The proposed OGSA-based approach is also applied for the minimization of TVD of this test power network. The results yielded by the proposed OGSA-based approach are presented in Table 3. The results obtained by the proposed OGSA are compared to those reported in the literature like GSA [13] and DE [9]. From this table, 5.37% improvement in TVD may be recorded by using the proposed OGSA-based algorithm as compared to GSA counterpart reported in [13]. Comparative DE-, GSA- and OGSA-based convergence profile of TVD (p.u.) for this test power system is presented in Fig. 3. From this figure it may be observed that the convergence profile of TVD (p.u.) for the proposed OGSA-based approach for this test system is promising one.

4.1.3 Minimization of System VSI for IEEE 30-Bus Power System

Results obtained by the proposed OGSA algorithm for the improvement of VSI for IEEE 30-bus test power system are also presented in Table 3. The results obtained by the proposed OGSA are compared to those reported in the literature like GSA [13] and DE [9]. From this table, 10.11% improvement in VSI may be recorded by using the proposed OGSA-based algorithm as compared to GSA reported in [13].

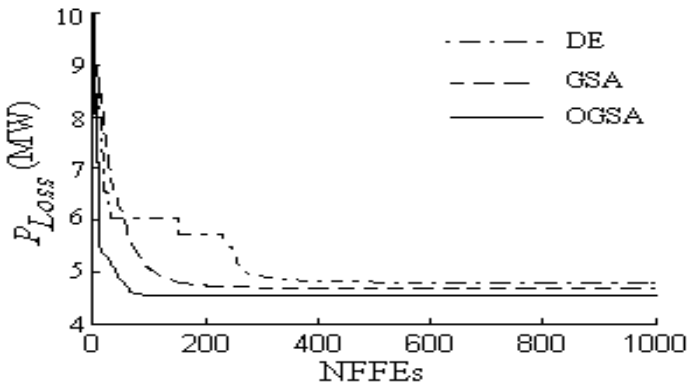


Fig. 2. Comparative convergence profiles of transmission loss for IEEE 30-bus test power system

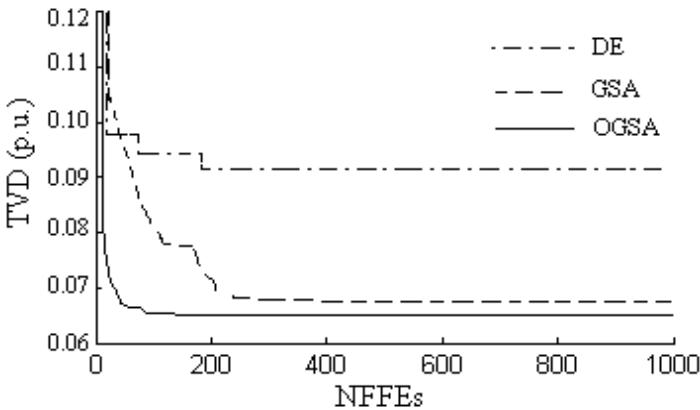


Fig. 3. Comparative convergence profile of TVD for IEEE 30-bus test power system

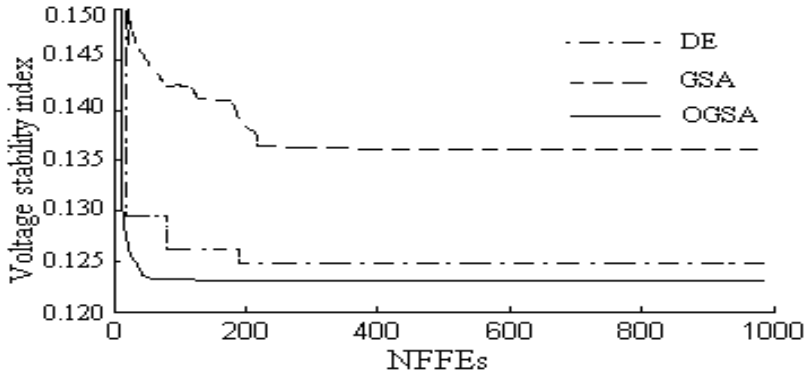


Fig. 4. Comparative convergence profile of VSI for IEEE 30-bus test power

5 Conclusion

In this paper, the proposed OGSA algorithm has been successfully implemented to solve ORPD problem of IEEE 30-bus test power network. It has been observed that the proposed OGSA algorithm has the ability to converge to a better quality near-optimal solution. This algorithm possesses better convergence characteristic and robustness than other prevailing techniques reported in the recent literature. It is also clear that the proposed OGSA algorithm is free from the shortcoming of premature convergence exhibited by the other optimization algorithms. Thus, the proposed algorithm may become a promising tool for solving some other more complex engineering optimization problems for the future researchers.

References

1. Durairaj, S., Kannan, P.S., Devaraj, D.: Multi-objective VAR dispatch using particle swarm optimization. *Emerg. Electr. Power Syst.* 4(1) (2005) ISSN(online) 1553-779X
2. Durairaj, S., Devaraj, D., Kannan, P.S.: Genetic algorithm applications to optimal reactive power dispatch with voltage stability enhancement. *IE (I) Journal-EL* 87, 42–47 (2006)
3. Devaraj, D., Durairaj, S., Kannan, P.S.: Real parameter genetic algorithm to multi-objective reactive power dispatch. *Int. J. Power Energy Syst.* 28(1) (2008) ISSN (online) 1710-2243
4. Wu, Q.H., Cao, Y.J., Wen, J.Y.: Optimal reactive power dispatch using an adaptive genetic algorithm. *Int. J. Electr. Power Energy Syst.* 20(8), 563–569 (1998)
5. Wu, Q.H., Ma, J.T.: Power system optimal reactive power dispatch using evolutionary programming. *IEEE Trans. Power Syst.* 10(3), 1243–1249 (1995)
6. Yoshida, H., Kawata, K., Fukuyama, Y., Takamura, S., Nakanishi, Y.: A particle swarm optimization for reactive power and voltage control considering voltage security assessment. *IEEE Trans. Power Syst.* 15(4), 1232–1239 (2000)
7. Esmin, A.A.A., Lambert-Torres, G., de-Souza, A.C.Z.: A hybrid particle swarm optimization applied to loss power minimization. *IEEE Trans. Power Syst.* 20(2), 859–866 (2005)

8. Tripathy, M., Mishra, S.: Bacteria foraging-based solution to optimize both real power loss and voltage stability limit. *IEEE Trans. Power Syst.* 22(1), 240–248 (2007)
9. Ela, A.A.A.E., Abido, M.A., Spea, S.R.: Differential evolution algorithm for optimal reactive power dispatch. *Electr. Power Syst. Res.* 81, 458–464 (2011)
10. Liang, C.H., Chung, C.Y., Wong, K.P., Duan, X.Z., Tse, C.T.: Study of differential evolution for optimal reactive power flow. *IEE Proc. Gener. Trans. Distrib.* 1(2), 253–260 (2007)
11. Varadarajan, M., Swarup, K.S.: Network loss minimization with voltage security using differential evolution. *Electr. Power Syst. Res.* 78, 815–823 (2008)
12. Dai, C., Chen, W., Zhu, Y., Zhang, X.: Seeker optimization algorithm for optimal reactive power dispatch. *IEEE Trans. Power Syst.* 24(3), 1218–1231 (2009)
13. Duman, S., Sönmez, Y., Güvenç, U., Yörükeren, N.: Optimal reactive power dispatch using a gravitational search algorithm. *IET Gener. Transm. Distrib.* 6(6), 563–576 (2012)
14. Chatterjee, A., Ghoshal, S.P., Mukherjee, V.: A maiden application of gravitational search algorithm with wavelet mutation for the solution of economic load dispatch problems. *Int. J. Bio-inspired Comput.* 4(1), 33–46 (2012)
15. Pal, K., Saha, C., Das, S., Coello, C.A.: Dynamic Constrained Optimization with Offspring Repair based Gravitational Search Algorithm. In: *IEEE Congress on Evolutionary Computation (CEC 2013)*, Cancún, México, pp. 2414–2421 (2013)
16. Shaw, B., Mukherjee, V., Ghoshal, S.P.: A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems. *Int. J. Elect. Power Energy Syst.* 35(1), 21–33 (2012)
17. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. *Information Sciences* 179(13), 2232–2248 (2009)
18. Tizhoosh, H.R.: Opposition-based learning: a new scheme for machine intelligence. In: *Proc. Int. Conf. Computational Intel, Modeling, Control and Automation*, pp. 695–701 (2005)
19. Wang, X.F., Song, Y., Irving, M.: *Modern power systems analysis*. Springer, New York (2008)
20. Bhattacharya, A., Chattopadhyay, P.K.: Solution of optimal reactive power flow using biogeography-based optimization. *Int. J. Electr. Electron Eng.* 4(8), 568–576 (2010)
21. Mahadevan, K., Kannan, P.S.: Comprehensive learning particle swarm optimization for reactive power dispatch. *Appl. Soft. Comput.* 10, 641–652 (2010)
22. Subbaraj, P., Rajnarayanan, P.N.: Optimal reactive power dispatch using self-adaptive real coded genetic algorithm. *Electr. Power Syst. Res.* 79, 374–381 (2009)

A Novel Swarm Intelligence Based Gravitational Search Algorithm for Combined Economic and Emission Dispatch Problems

Hari Mohan Dubey¹, Manjaree Pandit¹, Bijaya Ketan Panigrahi², and Mugdha Udgir¹

¹ Department of Electrical Engineering,

Madhav Institute of Technology & science Gwalior, India

² Department of Electrical Engineering, Indian Institute of Technology Delhi, India

harimohandubey@rediffmail.com, manjaree_p@hotmail.com,
bkpanigrahi@ee.iitd.ac.in, mugdhaudgir@gmail.com

Abstract. In this article swarm Intelligence based gravitational search algorithm (PSOGSA) is used to solve combined economic and emission dispatch (CEED) problems. The CEED problem is modeled with the objective of minimizing fuel cost as well as emission level while satisfying associated operational constraints. Here the multi-objective function is converted into single objective function using price penalty method. The performance of PSOGSA approach is investigated on standard 10 unit system, 6 unit system and 40 unit system. The results obtained by simulation are compared with the recent reported results. The simulation result shows the fast convergence and its potential to solve complicated problems in power system.

1 Introduction

The aim of classical economic load dispatch (ELD) problem is to determine the schedule of generation which minimizes cost while satisfying load demand and operational constraints [1]. However this single objective problem is no longer considered alone due to the increased concern about the pollution and its harmful effects. Moreover the combustion of fossil fuel in thermal power plant releases NO_x, SO_x and CO_x emissions. These emissions have adverse effect on human lives as well as on environment. This led utilities to adopt modified design to minimize the pollutants. Also cost and emission are the two conflicting objectives which cannot be minimized simultaneously and generates a set of solution which cannot be considered optimal solution to the problem. Thus a compromise solution between the fuel cost and emission was obtained with the variation in weights. This bi-objective problem is converted into a single objective using price penalty factor method and is known as combined economic and emission dispatch problem (CEED).

Conventional optimization method such as goal programming technique [2], weighted min-max method [3] were used to solve the multi-objective optimization problem. Other evolutionary computation algorithms such as genetic algorithm(GA) [4], differential evolution (DE)[5], particle swarm optimization (PSO)[6], gravitational search algorithm (GSA)[7], chaotic ant swarm optimization (CASO)[8]

has been applied successfully to solve the EED problems. Gravitational search algorithm with wavelet mutation (GSAWM) [14] is also applied to solve the complex ELD problem.

Recently numerous techniques like multi-objective harmony search (MOHS) [9], oppositional gravitational search algorithm (OGSA) [10], multi-objective differential evolution algorithm (MODE) [11], fuzzy based bacterial foraging algorithm (MBFA) [12], quasi-oppositional teaching learning based optimization (QOTLBO) [13], are applied to solve these problem.

The PSO-GSA [16] technique is a hybrid evolutionary algorithm which incorporates the power of PSO in searching the global optimal solution into GSA with a higher convergence speed and accuracy. The PSO-GSA algorithm modifies the velocity of GSA using the global search capability of PSO such that the convergence speed is improved.

This article presents the application of PSO-GSA algorithm to solve the CEED problem with fuzzy decision scheme. To measure the effectiveness of hybrid PSO-GSA algorithm, it has been implemented on three standard test cases with different dimension and complexity. The rest of article is organized as follows: mathematical model of CEED problem is described in section 2. A brief description of PSO-GSA is explained in section 3. The simulation study and the numerical results obtained are presented in section 4. Finally, conclusion is drawn in section 5.

2 Mathematical Model of CEED Problem

2.1 Economic Load Dispatch Problem

The objective function for ELD is modeled as:

$$\min f = \sum_{i=1}^{N_G} F_i(P_i) \tag{1}$$

Where P_i and F_i are real power output (MW) and total cost of i^{th} generating unit, a_i , b_i and c_i are the cost coefficients of i^{th} generating unit. N_G is the number of generating units.

$$\text{Where } F_i(P_i) = a_i P_i^2 + b_i P_i + c_i \tag{2}$$

Considering valve point loading effect, cost function is given by-

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i + |e_i \times \sin(f_i \times (P_i^{min} - P_i))| \tag{3}$$

Where e_i and f_i are the cost coefficients for valve point loading effects.

2.2 Emission Dispatch

The aim of emission dispatch is to minimize the atmospheric pollutants caused by the combustion of fossil fuel in thermal power plants. Mathematically the function is expressed as:

$$E_i(P_i) = \sum_{i=1}^{N_G} \alpha_i P_i^2 + \beta_i P_i + \gamma_i + \eta_i \exp(\delta_i P_i) \tag{4}$$

Where $\alpha_i, \beta_i, \gamma_i, \eta_i$ and δ_i are the emission coefficients corresponding to i^{th} generating unit.

The generation cost and emission are minimized subjected to the following constraints:

1) *Generator capacity constraints-*

To obtain stable operation output power of each generating unit must lie within its lower P_i^{min} and upper P_i^{max} limits-

$$P_i^{min} \leq P_i \leq P_i^{max} \quad (5)$$

2) *Load balance constraints-*

$$\sum_{i=1}^{N_G} P_i = P_D + P_L \quad (6)$$

Where P_D system load demand and P_L is the real power transmission loss. P_L is calculated using B - coefficients, expressed as-

$$P_L = \sum_{i=1}^{N_G} \sum_{j=1}^{N_G} P_i B_{ij} P_j \quad (7)$$

2.3 Combined Economic and Emission Dispatch Problem

The two competing objectives, cost and emission are combined together using price penalty factor to obtain a single objective function. The fuel cost is calculated in \$/hr and emission is defined in lb/hr but for a single objective function it is necessary to measure both in same units, for this a price penalty factor is introduced with emission so that the total cost is obtained in \$/hr. The CEED problem minimizes the cost and emission simultaneously while satisfying the load demand and associated constraints. The problem can be formulated as-

$$\min F_{CEED} = u \times F_i(P_i) + (1 - u) \times h_i \times E_i(P_i) \quad (8)$$

Where F_{CEED} is the total cost in \$/hr, $0 < u < 1$ is a compromise factor which is a random number between the interval [0, 1] and h_i is the price penalty factor in \$/hr and is calculated as-

$$h_i = \frac{F_i(P_i^{max})}{E_i(P_i^{max})} \quad (9)$$

When u is 1, the function becomes the classical ELD problem. When u is 0, the function becomes only emission dispatch function minimizing the emission of the system. Therefore to obtain a compromise solution between cost and emission, value of u should lie somewhere between 0 and 1.

2.4 Best Compromise Solution

To obtain the best compromise solution from a set of non dominated solutions, fuzzy based mechanism is used. In this the membership value of every objective from the Pareto front is computed using the membership function defined as [15]:

$$\mu(F_i) = \begin{cases} 1; & F_i \leq F_i^{min} \\ \frac{F_i^{max}-F_i}{F_i^{max}-F_i^{min}}; & F_i^{min} < F_i < F_i^{max} \\ 0; & F_i \geq F_i^{max} \end{cases} \quad (10)$$

Where F_i^{min} and F_i^{max} are the minimum and maximum values of the i^{th} objective. The normalized membership function μ^k for each non-dominated solution k is evaluated as:

$$\mu^k = \frac{\sum_{i=1}^{N_{obj}} \mu_i^k}{\sum_{k=1}^M \sum_{i=1}^{N_{obj}} \mu_i^k} \quad (11)$$

Where N_{obj} represents the number of objectives, M is the obtained non-dominated solutions from the Pareto front. The best compromise solution is the one for which μ^k is maximum.

3 Hybrid PSO GSA

The PSO GSA is a heuristic optimization algorithm described by S. Mirjalili et al. in 2010[16]. Hybridization involves the combination of functionality of two algorithms i.e. it uses gbest of PSO in calculating the global best position with that of local search ability of GSA. This retains the best position found so far on the search space so that the particle having the best fitness value moves more slowly than the particle having lower fitness value.

In this algorithm, initially a population of N agents is assumed and is determined randomly in the search space.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n), \quad i = 1, 2, \dots, N \quad (12)$$

Where x_i^d corresponds to the i^{th} position of agent in dimension d , n represents the dimension of the problem.

The force at particular time t acting on mass i from mass j is expressed as follows:

$$F_{ij}^d(t) = G(t) \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \epsilon} (x_j^d(t) - x_i^d(t)) \quad (13)$$

Where $M_i(t)$ and $M_j(t)$ are masses of agents i and j , $G(t)$ represents the gravitational constant at time t , $R_{ij}(t)$ is the Euclidean distance between the two agents i and j and ϵ is a small constant.

$$R_{ij}(t) = \| X_i(t), X_j(t) \|_2 \quad (14)$$

Gravitational constant $G(t)$ is calculated as:

$$G(t) = G_0 e^{-\alpha \frac{t}{T}} \quad (15)$$

Where G_0 is expressed as the initial value of gravitational constant, α is user specified constant, t is the current iteration and T corresponds to maximum number of iterations.

The acceleration of agent i in dimension d at time t is expressed as follows:

$$\alpha_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \quad (16)$$

The total force on agent i in dimension d is calculated as follows:

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}^d(t) \quad (17)$$

Where $rand_j$ is the random number determined in the interval $[0, 1]$.

In each iteration, the velocity of the next agent is described as-

$$v_i^d(t+1) = w \cdot v_i^d(t) + c_1' \times rand \times \alpha_i^d(t) + c_2' \times rand \times (gbest - x_i^d(t)) \quad (18)$$

Where $v_i^d(t)$ is the velocity of agent i at iteration t in dimension d , w is a weighting function, c_j' is a weighting factor, $rand$ is a random number between 0 and 1, $\alpha_i^d(t)$ is the acceleration of agent i at iteration t in dimension d , $gbest$ is the best position of swarm found so far and x_i^d is the position of agent i in d^{th} dimension.

The next position of agent is calculated as-

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (19)$$

Where $v_i^d(t+1)$ is the velocity of next agent and x_i^d is the position of i^{th} agent in dimension d at iteration t .

With the evaluation of fitness value, masses can be updated as follows:

$$m_i(t) = \frac{currentfitness_i(t) - 0.99 * worst(t)}{best(t) - worst(t)} \quad i = 1, 2, \dots, N \quad (20)$$

$$M_i(t) = \frac{m_i(t) * 5}{\sum_{j=1}^N m_j(t)} \quad (21)$$

Where $currentfitness_i(t)$ corresponds to the fitness value of the agent i at any time t , and $best(t)$ and $worst(t)$ are the minimum and maximum fitness value of all agents.

4 Numerical Example and Simulation Results

The effectiveness of hybrid PSO-GSA for the solution of CEED problem is investigated on three standard test systems. These are 10 unit with non smooth cost function and loss coefficient, 6 unit system with smooth cost function and loss coefficients and 40 unit system with non-smooth cost function and without loss coefficients. The algorithm is implemented in MATLAB 7.8 and system configuration is Intel core i3 processor with 2.3 GHz and 2 GB RAM.

4.1 Test System I: Ten Unit System

This test system contains ten thermal generating unit with valve point loading effect in cost and emission function. In this system loss coefficients are also considered. The entire system data is adopted from [7], with a demand of 2000MW. The results obtained in terms of optimum power output, cost, emission, transmission loss and average computational time by different methods is presented in table 1. The best

value in terms of cost achieved by PSO-GSA approach is 113459.8719 \$/hr and in terms of emission is 4110.24916 lb/hr. Comparison of results is made with strength Pareto evolutionary algorithm 2(SPEA 2)[11], non dominated sorting genetic algorithm II(NSGA-II) [11], Pareto differential evolution (PDE) [11], multi-objective differential evolution algorithm (MODE) [11], gravitational search algorithm (GSA) [7], quasi-oppositional teaching learning based optimization (QOTLBO)[13] and teaching learning based optimization(TLBO)[13]. Table 1, shows that the compromise solution obtained from the hybrid PSO-GSA approach is less than the other reported methods. The Pareto optimal front obtained by hybrid PSO-GSA for 10 unit system is presented in Fig.1.

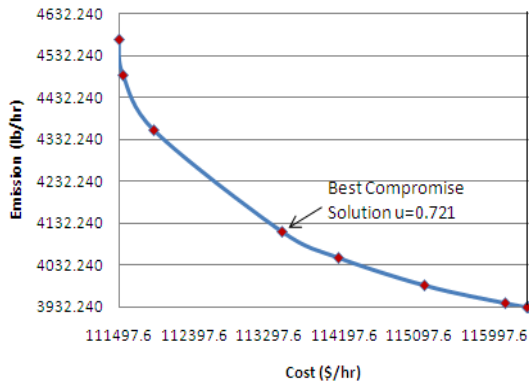


Fig. 1. Pareto optimal front for ten unit system

Table 1. Result of 10 unit system with a demand of 2000 MW

| Unit | PSOGSA | QOLTBO [13] | TLBO [13] | GSA [7] | MODE [11] | PDE [11] | NSGA-II [11] | SPEA 2 [11] |
|-------------------------|--------------------|-------------|-----------|----------|-----------|----------|--------------|-------------|
| Pg1 | 55.0000 | 55.0000 | 55.0000 | 54.9992 | 54.9487 | 54.9853 | 51.9515 | 52.9761 |
| Pg2 | 80.0000 | 80.0000 | 80.0000 | 79.9586 | 74.5821 | 79.3803 | 67.2584 | 72.8130 |
| Pg3 | 84.9001 | 84.8457 | 83.9202 | 79.4341 | 79.4294 | 83.9842 | 73.6879 | 78.1128 |
| Pg4 | 83.9537 | 83.4993 | 82.8342 | 85.0000 | 80.6875 | 86.5942 | 91.3554 | 83.6088 |
| Pg5 | 142.7178 | 142.9210 | 132.0131 | 142.1063 | 136.8551 | 144.4386 | 134.0522 | 137.2432 |
| Pg6 | 163.4025 | 163.2711 | 173.9880 | 166.5670 | 172.6393 | 165.7756 | 174.9504 | 172.9188 |
| Pg7 | 299.8566 | 299.8066 | 299.7099 | 292.8749 | 283.8233 | 283.2122 | 289.4350 | 287.2023 |
| Pg8 | 315.4092 | 315.4388 | 317.9684 | 313.2387 | 316.3407 | 312.7709 | 314.0556 | 326.4023 |
| Pg9 | 428.4821 | 428.5084 | 427.0166 | 441.1775 | 448.5923 | 440.1135 | 455.6978 | 448.8814 |
| Pg10 | 430.1033 | 430.5524 | 431.3955 | 428.6306 | 436.4287 | 432.6783 | 431.8054 | 423.9025 |
| Cost (\$/hr) | 113459.8719 | 113460 | 113471 | 113490 | 113480 | 113510 | 113540 | 113520 |
| Emission (lb/hr) | 4110.24916 | 4110.2 | 4113.5 | 4111.4 | 4124.9 | 4111.4 | 4130.2 | 4109.1 |
| PLoss (MW) | 83.8253 | 83.8433 | 83.8459 | NA | NA | NA | NA | NA |

4.2 Test System II: Six Unit System

For this test case the unit data for cost, emission and losses is adapted from[11], the load demand is set at 1200 MW. The best value in terms of cost achieved by PSO-GSA method is 64911.6638 \$/hr and the best value in terms of emission is 1281.44834 lb/hr. Comparison of result in terms of generator schedule, cost, emission

and losses were made with quasi-oppositional teaching learning based optimization QOTLBO[13], teaching learning based optimization(TLBO)[13], strength Pareto evolutionary algorithm 2(SPEA 2)[11], non dominated sorting genetic algorithm II(NSGA-II) [11], Pareto differential evolution(PDE) [11], multi-objective differential evolution algorithm(MODE) [11] in table 2. The Pareto optimal front obtained by the hybrid PSO/GSA approach for 6 unit system is shown in Fig. 2.

Table 2. Result of six unit system with a demand of 1200 MW

| Unit | PSOGSA | QOTLBO [13] | TLBO [13] | MODE [11] | PDE [11] | NSGA-II [11] | SPEA2 [11] |
|-------------------------|-------------------|-------------|-----------|-----------|----------|--------------|------------|
| Pg1 | 107.4170 | 107.3101 | 107.8651 | 108.6284 | 107.3965 | 113.1259 | 104.1573 |
| Pg2 | 121.4823 | 121.4970 | 121.5676 | 115.9456 | 122.1418 | 116.4488 | 122.9807 |
| Pg3 | 206.3238 | 206.5010 | 206.1771 | 206.7969 | 206.7536 | 217.4191 | 214.9553 |
| Pg4 | 205.4595 | 206.5826 | 205.1879 | 210.0000 | 203.7047 | 207.9492 | 203.1387 |
| Pg5 | 307.1902 | 304.9838 | 306.5555 | 301.8884 | 308.1045 | 304.6641 | 316.0302 |
| Pg6 | 303.6222 | 304.6036 | 304.1423 | 308.4127 | 303.3797 | 291.5969 | 289.9396 |
| Cost (\$/hr) | 64911.6638 | 64912 | 64922 | 64843 | 64920 | 64962 | 64884 |
| Emission (lb/hr) | 1281.44834 | 1281 | 1281 | 1286 | 1281 | 1281 | 1285 |
| PLoss (MW) | 51.4949 | 51.4955 | 51.4955 | NA | NA | NA | NA |

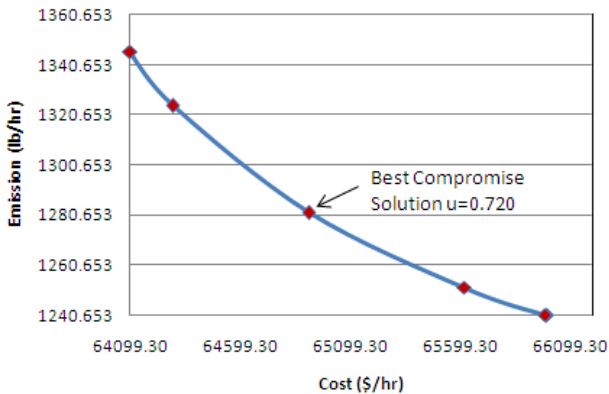


Fig. 2. Pareto optimal front for six unit system

4.3 Test System III: Forty Unit System

A 40 generating unit system with valve point loading effect in the fuel cost and emission functions with loss coefficients. The load demand is set as 10500 MW. The system data is taken from [7]. The best value of cost achieved by PSOGSA is 125681.1656 \$/hr and emission is 197532.94859 lb/hr. Table 3 shows the comparison of result obtained by PSOGSA with quasi-oppositional based teaching learning based optimization(QOTLBO)[13], teaching learning based optimization(TLBO)[13], non dominated sorting genetic algorithm II(NSGA-II) [11], Pareto differential evolution(PDE) [11], multi objective differential evolution algorithm(MODE) [11]

and gravitational search algorithm(GSA) [7]. Solution obtained by PSO GSA in terms of fuel cost is less than the other reported methods except QOTLBO [13] and TLBO [13]; however the emission level obtained by PSO GSA method is less as compared to all methods. Fig. 3 shows the Pareto optimal front obtained by PSO GSA approach for 40 unit system.

Table 3. Result of 40 unit system with a demand of 10500 MW

| Unit | PSOGSA | QOTLBO [13] | TLBO [13] | GSA [7] | MODE [11] | PDE [11] | NSGA-II [11] |
|-------------------------|---------------------|-------------|-----------|----------|-----------|----------|--------------|
| Pg1 | 114.0000 | 114.0000 | 114.0000 | 113.9989 | 113.5295 | 112.1549 | 113.8685 |
| Pg2 | 114.0000 | 114.0000 | 114.0000 | 113.9896 | 114.0000 | 113.9431 | 113.6381 |
| Pg3 | 120.0000 | 120.0000 | 91.9893 | 119.9995 | 120.0000 | 120.0000 | 120.0000 |
| Pg4 | 179.7331 | 179.7593 | 177.4467 | 179.7857 | 179.8015 | 180.2647 | 180.7887 |
| Pg5 | 97.0000 | 97.0000 | 97.0000 | 97.0000 | 96.7716 | 97.0000 | 97.0000 |
| Pg6 | 140.0000 | 140.0000 | 140.0000 | 139.0128 | 139.2760 | 140.0000 | 140.0000 |
| Pg7 | 284.2133 | 300.0000 | 300.0000 | 299.9885 | 300.0000 | 299.8829 | 300.0000 |
| Pg8 | 290.1226 | 298.9093 | 283.7368 | 300.0000 | 298.9193 | 300.0000 | 299.0084 |
| Pg9 | 285.4145 | 300.0000 | 300.0000 | 296.2025 | 290.7737 | 289.8915 | 288.8890 |
| Pg10 | 200.3682 | 130.0996 | 130.0000 | 130.3850 | 130.9025 | 130.5725 | 131.6132 |
| Pg11 | 318.2831 | 243.7055 | 318.1965 | 245.4775 | 244.7349 | 244.1003 | 246.5128 |
| Pg12 | 316.3484 | 318.4741 | 241.5727 | 318.2101 | 317.8218 | 318.2840 | 318.8748 |
| Pg13 | 394.2793 | 394.4004 | 391.9916 | 394.6257 | 395.3846 | 394.7833 | 395.7224 |
| Pg14 | 394.2794 | 394.3418 | 394.4501 | 395.2016 | 394.4692 | 394.2187 | 394.1369 |
| Pg15 | 394.2798 | 394.2703 | 394.3549 | 306.0014 | 305.8104 | 305.9616 | 305.5781 |
| Pg16 | 394.2799 | 394.4013 | 394.0597 | 395.1005 | 394.8229 | 394.1321 | 394.6968 |
| Pg17 | 486.2944 | 489.3143 | 490.5281 | 489.2569 | 487.9872 | 489.3040 | 489.4234 |
| Pg18 | 488.3961 | 489.3548 | 484.2089 | 488.7598 | 489.1751 | 489.6419 | 488.2701 |
| Pg19 | 421.5203 | 511.1648 | 423.9535 | 499.2320 | 500.5265 | 499.9835 | 500.8000 |
| Pg20 | 421.5206 | 421.8134 | 507.3859 | 455.2821 | 457.0072 | 455.4160 | 455.2006 |
| Pg21 | 433.5204 | 434.5654 | 438.5029 | 433.4520 | 434.6068 | 435.2845 | 434.6639 |
| Pg22 | 433.5215 | 434.5536 | 433.6163 | 433.8125 | 444.5310 | 433.7311 | 434.1500 |
| Pg23 | 433.5191 | 433.9734 | 434.1238 | 445.5136 | 444.6732 | 446.2496 | 445.8385 |
| Pg24 | 433.5211 | 433.7659 | 446.0748 | 452.0547 | 452.0332 | 451.8828 | 450.7509 |
| Pg25 | 433.5203 | 434.9881 | 437.2666 | 492.8864 | 492.7831 | 493.2259 | 491.2745 |
| Pg26 | 433.5205 | 434.1780 | 433.3886 | 433.3695 | 436.3347 | 434.7492 | 436.3418 |
| Pg27 | 10.0000 | 10.0574 | 10.2118 | 10.0026 | 10.0000 | 11.8064 | 11.2457 |
| Pg28 | 10.0000 | 10.3295 | 11.1608 | 10.0246 | 10.3901 | 10.7536 | 10.0000 |
| Pg29 | 10.0000 | 10.0147 | 10.2531 | 10.0125 | 12.3149 | 10.3053 | 12.0714 |
| Pg30 | 97.0000 | 97.0000 | 97.0000 | 96.9125 | 96.9050 | 97.0000 | 97.0000 |
| Pg31 | 186.0240 | 190.0000 | 190.0000 | 189.9689 | 189.7727 | 190.0000 | 189.4826 |
| Pg32 | 190.0000 | 190.0000 | 190.0000 | 175.0000 | 174.2324 | 175.3065 | 174.7971 |
| Pg33 | 190.0000 | 190.0000 | 190.0000 | 189.0181 | 190.0000 | 190.0000 | 189.2845 |
| Pg34 | 200.0000 | 200.0000 | 200.0000 | 200.0000 | 199.6506 | 200.0000 | 200.0000 |
| Pg35 | 200.0000 | 200.0000 | 200.0000 | 200.0000 | 199.8662 | 200.0000 | 199.9138 |
| Pg36 | 200.0000 | 200.0000 | 200.0000 | 199.9978 | 200.0000 | 200.0000 | 199.5066 |
| Pg37 | 110.0000 | 110.0000 | 110.0000 | 109.9969 | 110.0000 | 109.9412 | 108.3061 |
| Pg38 | 110.0000 | 110.0000 | 110.0000 | 109.0126 | 109.9454 | 109.8823 | 110.0000 |
| Pg39 | 110.0000 | 110.0000 | 110.0000 | 109.4560 | 108.1786 | 108.9686 | 109.7899 |
| Pg40 | 421.5206 | 421.5651 | 459.5306 | 421.9987 | 422.0682 | 421.3778 | 421.5609 |
| Cost (\$/hr) | 125681.1656 | 125161 | 125602 | 125782 | 125792 | 125731 | 125825 |
| Emission (lb/hr) | 197532.94859 | 206490.4 | 206648.3 | 210932.9 | 211190.2 | 211765.5 | 210949.1 |

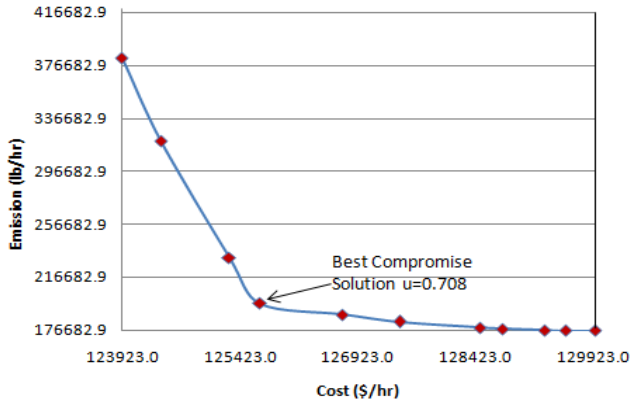


Fig. 3. Pareto optimal front for 40 unit system

4.4 Parameter Selection

In PSO-GSA, population (N), gravitational constant (G_0), acceleration coefficient (α), weighting factor (C1 and C2) are the four control parameters. The performance of algorithm is dependent on these parameters therefore a detailed study was carried out to obtain the optimal value by varying the parameters. Here 20 independent trials have been made for each combination and maximum numbers of iterations are set to 500. The performance of PSO-GSA is tested on 6 unit system with loss coefficients. After a careful experimentation the following input parameters were selected to obtain the optimal value: population size (N) =60, gravitational constant(G_0) =1, acceleration coefficient(α)=10, C1=2.5, C2=1.0 and maximum number of iteration=500.

1) *Effect of C1 and C2 on PSO-GSA algorithm*

Initially parameter set at N =60, G_0 =1 and α =10. To obtain the optimum value of weighting factor C1 and C2, the value of C1 and C2 were increased from 1.0 to 2.5 with a step size of 0.5. Table 4 shows the effect of variation of C1 and C2 for obtaining minimum, maximum, mean cost and standard deviation for 20 repeated trials. The optimal value which gives best results is: C1=2.5 and C2=1.0.

2) *Effect of acceleration coefficient on the PSO-GSA performance*

Change in acceleration coefficient affects the PSO-GSA performance. Increasing acceleration coefficient does not produce any significant improvement in the results rather it increases the standard deviation and computational speed of algorithm and is not capable of searching the minimum. Table 5 shows the performance of acceleration coefficient for different values keeping N =60, G_0 =1, C1=2.5 and C2=1.0.

3) *Effect of gravitational constant on PSO-GSA*

Too large value of gravitational constant makes the algorithm computationally inefficient, also the average cost, standard deviation and maximum cost increases with increases in the value of gravitational constant. G_0 is varied from 1 to 30 with a step size of 10. It can be seen from the Table 6 that minimum cost was obtained when G_0 =1.

Table 4. Effect of C1 and C2 on 6 unit system (test case II, 20 trials)

| Case | C1 | C2 | Min cost (\$/hr) | Avg cost (\$/hr) | Max cost (\$/hr) | SD |
|------|-----|------------|-------------------|-------------------|-------------------|---------------|
| 1. | 1.0 | 1.0 | 64914.2364 | 64919.1901 | 64923.1712 | 4.0044 |
| 2. | | 1.5 | 64914.9443 | 64917.6482 | 64923.8382 | 5.3749 |
| 3. | | 2.0 | 64918.1279 | 65302.2253 | 65499.6911 | 332.6822 |
| 4. | | 2.5 | 65233.3684 | 65294.6299 | 65377.7930 | 74.6617 |
| 5. | 1.5 | 1.0 | 64911.9147 | 64914.9325 | 64917.4635 | 2.8045 |
| 6. | | 1.5 | 64911.7101 | 64913.5514 | 64915.9932 | 2.2037 |
| 7. | | 2.0 | 65264.6206 | 65346.2945 | 65427.9689 | 115.5046 |
| 8. | | 2.5 | 64912.4194 | 65191.1590 | 65398.1173 | 250.6791 |
| 9. | 2.0 | 1.0 | 64913.5478 | 64915.3488 | 64917.8698 | 2.24914 |
| 10. | | 1.5 | 64914.5142 | 64915.4958 | 64916.4458 | 0.9661 |
| 11. | | 2.0 | 64913.5534 | 65025.5271 | 65248.2012 | 192.8424 |
| 12. | | 2.5 | 64914.3474 | 65444.3139 | 65974.2805 | 749.4858 |
| 13. | 2.5 | 1.0 | 64911.6638 | 64912.7645 | 64914.7399 | 1.3746 |
| 14. | | 1.5 | 64912.4963 | 64913.2655 | 64914.5602 | 1.1278 |
| 15. | | 2.0 | 64916.0428 | 65032.9721 | 65266.0821 | 201.8795 |
| 16. | | 2.5 | 64914.7834 | 64915.7069 | 64916.6304 | 1.3060 |

Table 5. Effect of α on 6 unit system (test case II, 20 trials)

| Sr. No. | | Min cost (\$/hr) | Avg cost (\$/hr) | Max cost (\$/hr) | SD |
|---------|----|------------------|------------------|------------------|--------|
| 1. | 10 | 64911.6638 | 64912.7645 | 64914.7399 | 1.3746 |
| 2. | 20 | 64912.2824 | 64919.8822 | 64928.1949 | 6.5481 |
| 3. | 30 | 64919.2769 | 64925.3023 | 64931.3277 | 8.5212 |

Table 6. Effect of G_0 on 6 unit system (test case II, 20 trials)

| Sr. No. | G_0 | Min cost (\$/hr) | Avg cost (\$/hr) | Max cost (\$/hr) | SD |
|---------|-------|------------------|------------------|------------------|---------|
| 1. | 1 | 64911.6638 | 64912.7645 | 64914.7399 | 1.3746 |
| 2. | 10 | 64914.7971 | 64915.8911 | 64916.9851 | 1.5471 |
| 3. | 20 | 64915.8571 | 64916.5560 | 64917.2550 | 0.98846 |
| 4. | 30 | 64912.6939 | 64931.2832 | 64949.8726 | 26.2893 |

4) *Effect of population size*

To achieve the global best results population size must be optimum. Large population makes the algorithm slow, increases average cost and standard deviation. Test was carried out with different population sizes 60,120,180 and 240 for 20 repeated trials. Table 7 list the performance of PSOGSA with different population size. From Table 7 it can be seen that N=60 gives best generation cost.

Table 7. Effect of population on 6 unit system (test case II, 20 trials)

| Sr. No. | N | Min cost (\$/hr) | Avg cost (\$/hr) | Max cost (\$/hr) | SD |
|---------|-----|------------------|------------------|------------------|---------|
| 1. | 60 | 64911.6638 | 64912.7645 | 64914.7399 | 1.3746 |
| 2. | 120 | 64911.6696 | 64913.4880 | 64914.6466 | 1.5944 |
| 3. | 180 | 64912.2418 | 64919.6968 | 64926.5717 | 7.1825 |
| 4. | 240 | 64913.3331 | 64914.3316 | 64915.3038 | 0.98561 |

4.5 Comparative Study

1) *Computational Efficiency*

As seen from the table 1, 2 and 3 the minimum cost achieved by the PSOGSA method is 113459.8719 \$/hr, 64911.6638 \$/hr and 125681.1656 \$/hr for test system I, II and

III respectively. The minimum cost obtained for test system III is 125681.1656 \$/hr is not best but the emission obtained is 197532.94859 ton/hr is less as compared to the reported methods in the literature. It can be seen from the table 8 that the minimum cost and average computational time obtained by PSO-GSA method is less as compared to other techniques. Therefore it can be concluded that PSO-GSA is computationally efficient than the other mentioned methods.

Table 8. Comparison of computational efficiency

| Test System | Method | Min cost (\$/hr) | Time/Iter (sec) |
|----------------|-------------|------------------|-----------------|
| 10 unit system | PSO-GSA | 113459.8719 | 0.00919 |
| | QOTLBO[13] | 113460 | 3.14 |
| | TLBO[13] | 113471 | 3.78 |
| | GSA[7] | 113490 | NA |
| | MODE[11] | 113480 | 3.82 |
| | PDE[11] | 113510 | 4.23 |
| | NSGA-II[11] | 113540 | 6.02 |
| | SPEA 2[11] | 113520 | 7.53 |
| 6 unit system | PSO-GSA | 64911.6638 | 0.00740 |
| | QOTLBO[13] | 64912 | 1.91 |
| | TLBO[13] | 64922 | 2.18 |
| | MODE[11] | 64843 | 3.09 |
| | PDE[11] | 64920 | 3.52 |
| | NSGA-II[11] | 64962 | 5.42 |
| | SPEA 2[11] | 64884 | 7.05 |
| | PSO-GSA | 125681.1656 | 0.0318 |
| 40 unit system | QOTLBO[13] | 125161 | 4.76 |
| | TLBO[13] | 125602 | 5.02 |
| | GSA[7] | 125782 | NA |
| | MODE[11] | 125792 | 5.39 |
| | PDE[11] | 125731 | 6.15 |
| | NSGA-II[11] | 125825 | 7.32 |

2) Solution Quality

It can be seen from the Table 1, 2 and 3 that the best compromise solution obtained from hybrid PSO-GSA method is less as compared to reported results. In this price penalty factor is multiplied with emission function so that equivalent cost curve is obtained in \$/hr. It is observed that PSO-GSA has the ability to achieve global minimum in consistent manner for different types of CEED problems.

5 Conclusion

In this article, the PSO-GSA algorithm is implemented on three standard CEED problem with different dimension and complexity. The CEED problem is formulated into a single objective function by employing price penalty factor approach, which aims to minimize the fuel cost and emission simultaneously. The obtained simulation result is compared with other approach available in recent literature, which depicts the efficiency of PSO-GSA in solving complex nonlinear problems. From the comparative study it can also be concluded that PSO-GSA has fast convergence and provides either better or comparable results than the other reported methods.

References

- [1] Wood, A.J., Wollenberg, B.F.: *Power Generation Operation and Control*. Wiley, New York (1984)
- [2] Nanda, J., Kothari, D.P., Linga Murthy, K.S.: Economic emission load dispatch through goal programming techniques. *IEEE Trans. on Energy Conversion* 3(1), 26–32 (1988)
- [3] Palanichamy, C., Babu, N.S.: Analytical solution for combined economic and emissions dispatch. *Electric Power System Research* 78, 1129–1137 (2008)
- [4] Guvenc, U.: Combined Economic Emission Dispatch solution using genetic algorithm based on similarity crossover. *Science Research and Essays* 5(17), 2451–2456 (2010)
- [5] Abou El Ela, A.A., Abido, M.A., Spea, S.R.: Differential evolution algorithm for emission constrained economic power dispatch problem. *Electric Power Systems Research* 80, 1286–1292 (2010)
- [6] Abido, M.A.: Multi-objective particle swarm optimization for environmental/economic dispatch problem. *Electric Power Systems Research* 79, 1105–1113 (2009)
- [7] Guvenc, U., Sonmez, Y., Duman, S., Yoruderen, N.: Combined Economic and Emission Dispatch solution using gravitational search algorithm. *Scientia Iranica* 19(6), 1754–1762 (2012)
- [8] Cai, J., Ma, X., Li, Q., Li, L., Peng, H.: A multi-objective chaotic ant swarm optimization for environmental/economic dispatch. *Electrical Power and Energy Systems* 32, 337–344 (2010)
- [9] Sivasubramani, S., Swarup, K.S.: Environmental/economic dispatch using multi-objective harmony search algorithm. *Electric Power Systems Research* 81, 1778–1785 (2011)
- [10] Shaw, B., Mukherjee, V., Ghoshal, S.P.: A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems. *Electrical Power and Energy Systems* 35, 21–33 (2012)
- [11] Basu, M.: Economic environmental dispatch using multi-objective differential evolution. *Applied Soft Computing* 11, 2845–2853 (2011)
- [12] 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)
- [13] Roy, P.K., Bhui, S.: Multi-objective quasi-oppositional teaching learning based optimization for economic emission load dispatch problem. *Electrical Power and Energy Systems* 53, 937–948 (2013)
- [14] Chatterjee, A., Ghoshal, S.P., Mukherjee, V.: A maiden application of gravitational search algorithm with wavelet mutation for the solution of economic load dispatch problems. *International Journal of Bio-inspired Computation* 4(1), 33–46 (2012)
- [15] Dhillon, J.S., Parti, S.C., Kothari, D.P.: Stochastic economic emission load dispatch. *Electric Power Syst. Res.* 26, 186–197 (1993)
- [16] Mirjalili, S., Mohd Hashim, S.Z.: A new Hybrid PSO-GSA Algorithm for Function Optimization. In: *IEEE International Conference on Computer and Information and Application (ICCIA)*, pp. 374–377 (2010)

Particle Swarm Optimization Based Optimal Reliability Design of Composite Electric Power System Using Non-sequential Monte Carlo Sampling and Generalized Regression Neural Network

R. Ashok Bakkiyaraj and Narayanan Kumarappan

Department of Electrical Engineering, Annamalai University, Chidambaram-608002, India
auashok@gmail.com, kumarappan_n@hotmail.com

Abstract. This paper presents a new approach based on particle swarm optimization (PSO) for determining the optimal reliability parameters of composite system using non-sequential Monte Carlo Simulation (MCS) and Generalized Regression Neural Network (GRNN). The cost-benefit based design model has been formulated as an optimization problem of minimizing system interruption cost and component investment cost. Solution of this design model requires the analysis of several reliability levels which needs to evaluate EDNS index for those levels. Evaluation of EDNS in non-sequential MCS requires state adequacy analysis for several thousands of sampled states. In conventional approaches, a dc load flow based load curtailment minimization model is solved for analyzing the adequacy of each sampled state which requires large computational resources. This paper reduces the computational burden by applying GRNN for state adequacy analysis of the sampled states. The effectiveness of the proposed methodology is tested on the IEEE 14-bus system.

Keywords: Composite electric power system, Reliability planning, Monte Carlo simulation, Generalized Regression Neural Network, Particle swarm optimization.

1 Introduction

The power utility satisfying the desired reliability can serve better to the customers than the system with sufficient reserve in normal demand condition. The objective of achieving better reliability of system components increases the energy price to the customers due to the increase in investment. But the improved reliability level of the utility reduces the interruption of power to the customers, thereby generates additional revenue for the power utility. The trade-off between the investment required for meet out the desired reliability and the reduction in power interruption cost can be the criteria for designing the optimum reliability parameter of system equipments [1-4].

Optimal design of components reliability parameters based on relative cost analysis are described in [5] and [6]. The optimization algorithms employed in these approaches are based on classical optimization techniques such as gradient projection method, polynomial-time algorithm and Bender's decomposition algorithm. Genetic

algorithm (GA) has used for reliability design of a power distribution system [7] and composite electric system by Su and Lii [8,9]. Design of optimal reliability parameters based on cost-benefit analysis systematically attempts to balance both the investment cost and the system interruption cost for better planning. It requires the evaluation of reliability indices for different system reliability levels. The states sampled in this optimization procedure are in the order of several millions and requires state adequacy evaluation for each state. Authors in their previous work developed a methodology based on non-sequential MCS for optimal reliability planning of composite electric power system. In that approach dc load flow based state adequacy model has solved using linear programming simplex method for several thousand sampled states of the same system [10].

The main objective of this paper is to develop a methodology for optimal reliability design of composite system based on Generalized Regression Neural Network (GRNN), non-sequential MCS and PSO algorithm. In this approach expected demand not served index (EDNS) for different reliability levels is evaluated using non-sequential MCS. The state adequacy analysis of the sampled states for a single reliability level is performed using dc load flow based load curtailment model and these analyzed states can be used as a training data of GRNN [12]. For other reliability levels of the same system, the sampled states are analyzed using the GRNN. This approach yields the optimal forced outage rate (U) parameters for system components and the system EDNS index. The validity of the proposed method is tested on IEEE 14-bus system [15]. The results obtained using GA is also provided for comparing the performance of the proposed method.

2 Methodology

In the era of smart grid, electric power utilities are under sustained pressure for providing power with adequate service reliability at lower energy tariff due to the competitive nature of supply industry. The design of optimal reliability parameters for system components are carried out after the system configuration and their power ratings are planned.

2.1 Optimal Reliability Design Model

In reliability planning the optimal value of forced outage rate of system components (U) are designed in such a way that to minimize the total interruption cost of the system plus the investment cost required for the components. The investment cost required for achieving the desired reliability parameters of each component is divided into installation cost and equipment manufacturing cost. Installation cost depends upon type of the component i.e. generator or transmission line, size of the element and location where components to be installed. The equipment cost involved in the manufacturing of reliable components depends on the design principles adopted, materials used and skilled workmanship etc. The cost involved in manufacturing the equipment is directly proportional to availability rate and inversely proportional to the outage rate. The investment cost given by [9] is

$$C_{investment} = \sum_{i=1}^n C_{instal}_i + C_{equip}_i \frac{1-U_i}{U_i} \quad (1)$$

Interruption cost is the revenue loss of the utility which depends on the shortage of energy supply to the customers due to the outages in generators and/or in transmission lines. This cost can be computed from the outage cost of 1 MWhr multiply by energy not supplied and is given by

$$C_{interrupt} = C_{loss} * EDNS * 8736 \quad (2)$$

The optimal reliability design model also imposing the limit on upper bounds of U and system expected demand not served (EDNS) to avoid the dilution of system reliability below certain level and can be defined as

$$\begin{aligned} \text{Min } C &= C_{investment} + C_{interrupt} \\ \text{Min } C &= \sum_{i=1}^n (C_{instal}_i + C_{equip}_i \frac{1-U_i}{U_i}) + C_{loss} * EDNS * 8736 \end{aligned} \quad (3)$$

Subject to

$$U_i \leq U_{imax} \quad (4)$$

$$EDNS \leq EDNS_{max} \quad (5)$$

Where C is the total cost, $C_{interrupt}$ is the system interruption cost, $C_{investment}$ is the total component investment cost, C_{loss} is the outage cost of 1MWhr, C_{instal}_i is the installation cost co-efficient for component i , C_{equip}_{iis} is the equipment cost coefficient for component i , U_i is the forced outage rate of component i , U_{imax} is the maximum value of forced outage rate of component I and $EDNS_{max}$ is the maximum value of system EDNS. This reliability design model is solved using the PSO algorithm in which each particle represents a potential solution i.e. each particle corresponds to a particular reliability level. The EDNS index evaluated based on non-sequential MCS and GRNN has used for calculating the interruption cost of each particle.

2.2 Evaluation of Expected Demand Not Served Index

The objective of calculating reliability indices in the non-sequential MCS approach is equivalent to calculating the expected value of a given test function $E(F)$ where $F(x)$ is the test function to verify whether the system state x is adequate. For EDNS index, the test function $F(x)$ represents the amount of load curtailment in MW required to alleviate the operating constraint violations and maintain power balance. For failure state load curtailment is a non-zero value ($F(x)>0$) and equal to zero for success state ($F(x)=0$). EDNS is evaluated by adding up all the unserved power demand over every sample divided by the total number of samples simulated. This work utilizes GRNN for state adequacy evaluation in MCS approach.

2.3 Generalized Regression Neural Network for State Adequacy Evaluation in non-Sequential MCS

The states sampled in the optimal reliability design optimization are in the order of several millions and requires estimation of test functions for each sampled state. Test functions can be estimated by conducting the state adequacy analysis using the dc

load flow based load curtailment minimization model [11] which requires the optimization procedure for each state. In this approach, computational requirement for state adequacy evaluation is reduced by developing a method based on GRNN. The training data for GRNN is obtained from the sampled states of MCS for the first particle of the initial generation. Each particle represents a particular system reliability level of the system. The adequacy analysis of the states sampled for first particle is performed using dc load flow based load curtailment minimization model. The trained GRNN is used for evaluate the adequacy of the states sampled for other particles and estimate the test functions for that reliability levels.

GRNN devised by Specht is based on non-linear regression theory for function estimation [12, 13]. It has the ability of performing kernel regression and non-parametric approximation of any arbitrary function between input and output vectors. The network architecture is a single pass learning algorithm with massive parallel structure. The topological structure of GRNN consists of input layer, pattern layer, summation layer and output layer as shown in Fig.1.

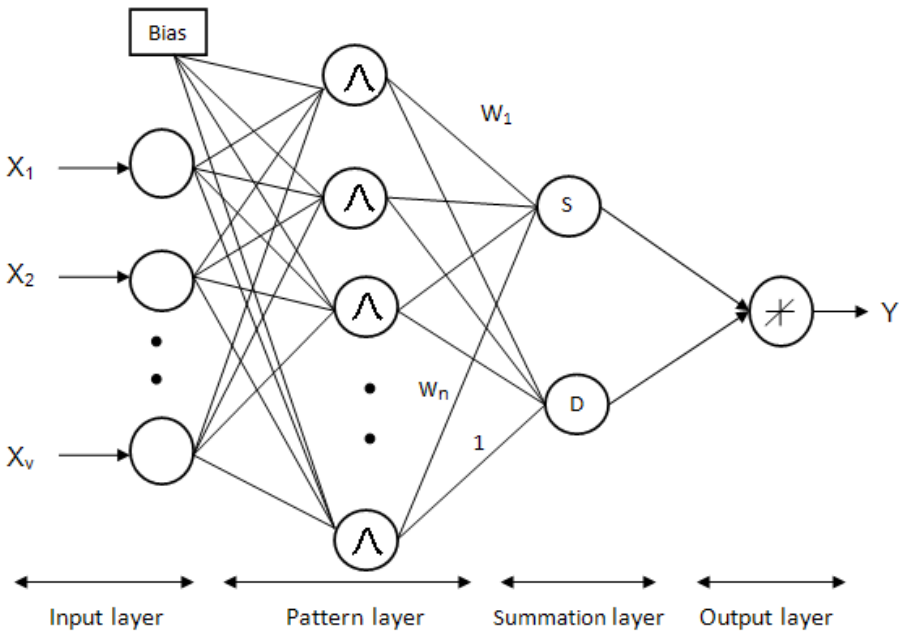


Fig. 1. GRNN architecture

The GRNN for state adequacy evaluation of sampled states has input layer neurons equal to number of system components and the output layer neurons equal to number of buses with loads. The number of hidden layer neurons is equal to number of training patterns in the data set which is set to ‘NS’. All hidden units simultaneously receive the n-dimensional binary valued input vector, where n is number of components in the system (i.e. 1 for up state and 0 for down state).

The input layer where no data processing performed is connected to the pattern layer. This pattern layer has a group of radial basis neurons and each neuron presents

a training pattern in the data set. Its output is a measure of the distance of the input from the stored pattern. Bias b is used to adjust the sensitivity of radial basis neuron, when established in the network it can be automatically set to $0.8326/\text{spread}$. Each neuron in the pattern layer is connected to two neurons in the summation layer. This layer has 'o' number of S-summation neurons and one D-summation neuron, where 'o' is the number of neurons in output layer. The S-summation neuron and D-summation neurons computes the sum of weighted outputs and unweighted outputs of the pattern layers respectively. The connection weights between pattern layer units and D-summation neuron is set to unity. The output layer neuron simply divides the output of each S-summation neuron by that of each D-summation neuron.

2.4 Implementation of PSO Algorithm for Optimal Reliability Design

PSO is a population based stochastic optimization technique developed by Kennedy and Eberhart in 1995 [14], inspired by social behavior of bird flocking or fish schooling. The optimal reliability planning model formulated as a non-linear optimization problem in equation (3) can be solved using the PSO algorithm as follows

Step 1:

Initialize the PSO parameters such as population size, maximum number of generations, number of variables, $C1$, $C2$, W_{max} , and W_{min} and read the problem parameters like number of generators, transmission lines, load bus, upper and lower limits on forced outage rate of system components and system EDNS, line data, bus data and sample size for MCS.

Step 2:

Set the forced outage rate of system components as unknown state variable $X = (x_1, x_2, \dots, x_n)$. Where n is total number of generators and transmission lines in the system.

Step 3:

Set the generation counter $t=0$ and generate initial population of 'p' particles randomly i.e. $\{ X^0(j) = (x_1, x_2, \dots, x_n), j = 1, \dots, p \}$. $X_i^0(j)$ is generated randomly by selecting a value with uniform probability over the search space $[X_{min}, X_{max}]$ using the eq.

$$X_i^0(j) = X_{min} + rand * (X_{max} - X_{min}) \quad (12)$$

Where X_{min} & X_{max} are minimum & maximum value of forced outage rate of that component, i is the component number and j is the particle number. Similarly generate random initial velocities of all particles $\{ V_0(j), j=1, \dots, p \}$ where $V_i^0(j)$ is generated by randomly selecting a value with uniform probability over the search space $[V_{min}, V_{max}]$.

Step 4:

If generation $t=0$ and particle $j=1$, for the first particle which represents a particular system reliability level, state adequacy evaluation is carried out for the states sampled in MCS using dc load flow based load curtailment model and estimate the test functions. A multi-dimensional array consists of components state and the value of the test function for all the samples is created. This is used as a training data of GRNN for state adequacy evaluation and proceeds to next step. Otherwise go to next step.

Step 5:

Evaluate the fitness of each particle according to the objective function given in equation (3) and check the constraint violations of each particle. This requires the calculation of EDNS index using non-sequential MCS with state adequacy evaluation using GRNN.

Step 6:

Form pbest set from each particle and assign gbest from the pbest set. Where pbest is the individual best of that particle and gbest is the global best of PSO algorithm.

Step 7:

Update the generation counter $t=t+1$.

Step 8:

Using the gbest and the pbest, update the j^{th} particle velocity using

$$V_j^{t+1} = W * V_j^t + C1 * \text{rand1} * (pbest_j - X_j^t) + C2 * \text{rand2} * (gbest - X_j^t).$$

Step 9:

Based on the updated velocities, each particle changes its positions as X_j^{t+1} . If the particle violates its position limits in any dimensions then set its position at the proper limit and do steps 5 and 6. Then go to next step.

Step 10:

When any of the stopping criteria is satisfied, like there is no change in the fitness value for certain number of consecutive generations or maximum number of generations reached, stop the algorithm or else go to step 6.

3 Numerical Results

The proposed optimal reliability design approach has been applied to IEEE 14-bus system. The investment cost co-efficient chosen in this simulation are similar to reference [10]. The performance of PSO for minimizing the optimal reliability design model has been compared with the results of GA. The results are not compared with reference [10], because the line data and maximum line loading of transmission lines are not given in that paper. PSO parameters fixed for simulation are population size is 20, maximum number of generation is 150, C1 is 1.8, C2 is 1.6, Wmax is 0.9 and Wmin is 0.4. GA parameters are population size is 20, maximum number of generations is 150, crossover rate is 0.85 and Mutation rate is 0.05. The optimization procedure is stopped when there is no change in the fitness value for 50 consecutive generations or when maximum number of generations reached, whichever occur first.

3.1 IEEE-14 Bus System

IEEE 14-bus system has 14 buses, 2 generators, 20 transmission lines and 11 bus loads [10, 15]. The maximum line flow ratings are assumed. The total load connected to the system is 259 MW and the outage cost assumed for loss of 1MWhr is 0.0000185 M\$. EDNS which is required for calculating the fitness for each particle is evaluated using non-sequential MCS with sample size of 10000. The GRNN for state adequacy evaluation has 22 neurons in the input layer and 11 neurons in the output layer with spread factor of 0.18. The training data for GRNN is obtained from the

50000 samples of the system reliability level represented by the first particle of the initial generation using dc load flow based load curtailment model. The upper bound of forced outage rate for generators is 0.05, transmission lines is 0.2 and the system EDNS is 35 MW.

Optimal forced outage rate parameters designed for generators and transmission lines of the system are presented in table 1. The PSO solution of the optimal design model converges to a total cost of 10.1562 M\$/year and GA results a total cost of 10.3954 M\$/year. PSO solution results a lower total cost than GA solution. It is inferred from the solutions that a higher investment reduces the system EDNS and achieves lower interruption cost and vice versa. This proves the applicability of the model used in optimal reliability design of system components. The convergence characteristics of PSO and GA for IEEE 14-bus system are given in fig. 2.

Table 1. Designed Optimal Forced Outage Rate Parameters for IEEE 14-bus System

| Component | GA | PSO |
|------------------------------|---------|---------|
| Generator 1 | 0.0393 | 0.0382 |
| Generator 2 | 0.0428 | 0.0481 |
| Line 1 | 0.0368 | 0.0214 |
| Line 2 | 0.0464 | 0.0422 |
| Line 3 | 0.0436 | 0.0242 |
| Line 4 | 0.0496 | 0.0745 |
| Line 5 | 0.0810 | 0.0471 |
| Line 6 | 0.0985 | 0.0963 |
| Line 7 | 0.0355 | 0.0473 |
| Line 8 | 0.0736 | 0.0311 |
| Line 9 | 0.0878 | 0.0505 |
| Line 10 | 0.1569 | 0.0952 |
| Line 11 | 0.1518 | 0.0983 |
| Line 12 | 0.0948 | 0.0354 |
| Line 13 | 0.1795 | 0.1655 |
| Line 14 | 0.0773 | 0.0506 |
| Line 15 | 0.1528 | 0.0502 |
| Line 16 | 0.1249 | 0.1153 |
| Line 17 | 0.1602 | 0.1208 |
| Line 18 | 0.1409 | 0.0522 |
| Line 19 | 0.1683 | 0.1910 |
| Line 20 | 0.1023 | 0.1819 |
| EDNS (MW) | 34.27 | 30.08 |
| Interruption cost (M\$/year) | 5.5385 | 4.8614 |
| Investment cost (M\$/year) | 4.8569 | 5.2948 |
| Total cost (M\$/year) | 10.3954 | 10.1562 |

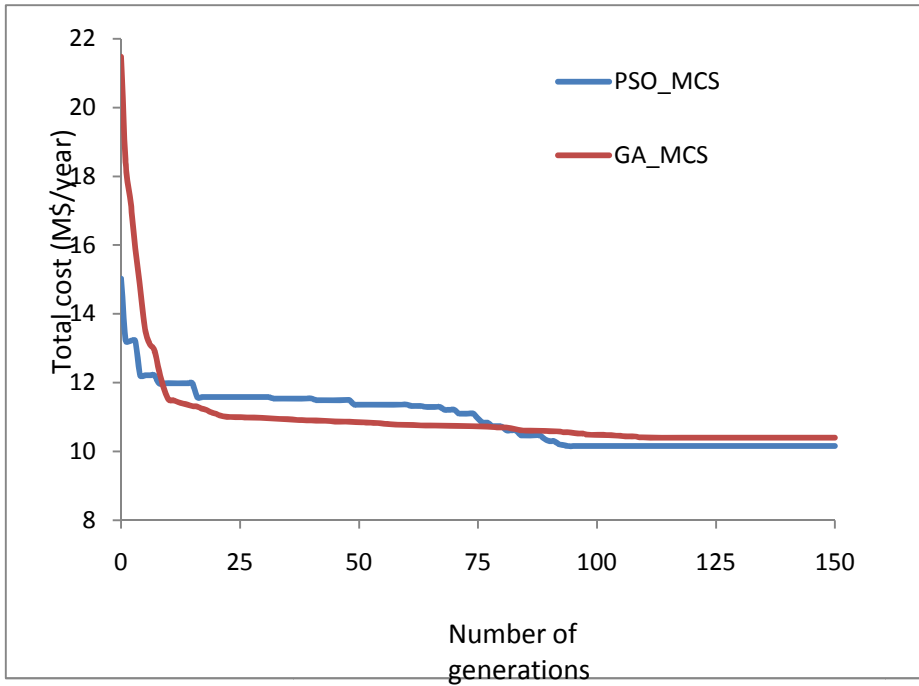


Fig. 2. Convergence Characteristics of IEEE 14-bus System

3.2 Discussion

The performance of the optimization technique in terms of the fitness value with PSO and GA is shown in figure 2. The result obtained in PSO and GA is summarized in table 1. For IEEE 14-bus system with EDNS evaluation using MCS, PSO method gives 5.2948 M\$/year as total investment cost where as GA method gives 4.8569 M\$/year, resulting in a lower system interruption cost of 4.8614 M\$/year in PSO method compared with the higher interruption cost of 5.5385 M\$/year given by GA method. PSO method gives a lower total cost of 10.1562 M\$/year where as GA method gives 10.3954 M\$/year. The PSO method converges to a minimum total cost than the GA method thereby resulting in a higher quality solution and also converges earlier than the GA method. Hence the PSO algorithm has better solution quality and convergence characteristics.

The system reliability is improved by more investment for system equipments and makes the system to be operated with less interruption and vice versa. This is reflected in the results discussed and, both PSO and GA achieve this objective. This proves the applicability of the design model for composite power system.

PSO requires 3000 (150*20, i.e maximum number of generation is 150 and number of particles is 20) and GA requires more than 3000 evaluations of EDNS index for calculating the interruption cost of the potential solutions generated. The states sampled for the evaluation of EDNS index for a particle in MCS is fixed as 10000 for

IEEE 14-bus system. The GRNN state adequacy evaluation approach eliminates several million optimization procedures required for solving the dc load flow based load curtailment model (for IEEE 14-bus system- 3000*10000). Thereby it drastically reduces the computational requirement of state adequacy evaluation in optimal reliability design. . Finally this proposed approach achieved the compromised solution and better total cost. Therefore the proposed method is cost effective and efficient.

4 Conclusion

A PSO based optimal design of reliability parameters for composite power system components such as generators and lines using non-Sequential MCS with GRNN based state adequacy evaluation has been presented. The optimal reliability design model has formulated in terms of components forced outage rates and system EDNS. Non-sequential MCS with fixed sample size are used for evaluating the EDNS index which is required for calculating the fitness of each particle. State adequacy evaluation for each sampled state in MCS is carried out using the GRNN for calculating the test function. By utilizing GRNN, several million numbers of optimization procedure required for the solution of dc load flow based load curtailment model has been eliminated. The feasibility of the proposed PSO approach has been verified by applying the present method on IEEE14-bus system. By analyzing the results it can be observed that optimal design of reliability parameters using PSO method gives higher quality solution by converge to a lower total cost and have better convergence characteristics than GA method. The proposed approach achieved the compromised solution with lower total cost than GA method. This proves the applicability of the design model selected for composite power system. So the proposed approach is cost effective and efficient, and can be used for modifying the reliability parameters of the existing systems and also in designing of reliability parameters for the new systems.

References

1. Neudorf, E.G., Kiguel, D.L., Hamoud, G.A., Porretta, B., Stephenson, W.M., Sparks, R.W., Logan, D.M., Bhavaraju, M.P., Billinton, R., Garrison, D.L.: Cost-benefit analysis of power system reliability: two utility case studies. *IEEE Transactions on Power Systems*, 1667–1675 (1995)
2. Billinton, R., Wacker, G., Wojczynski, E.: Comprehensive bibliography on electric service interruption costs. *IEEE TPAS*, 1831–1837 (1983)
3. Tollefson, G., Billinton, R., Wacker, G.: Comprehensive bibliography on reliability worth and electrical service consumer interruption costs: 1980–1990. *IEEE Transactions on Power Systems*, 1508–1514 (1991)
4. Billinton, R., Zhang, W.: Cost related reliability evaluation of bulk power systems. *Electric Power Systems Research*, 99–112 (2001)
5. Sallam, A., Desouky, M., Desouky, H.: Evaluation of optimal reliability indices fo electrical distribution systems. *IEEE Transactions on Reliability*, 259–264 (1990)
6. Chang, W.-F., Wu, Y.-C.: Optimal reliability design in an electrical distribution system via a polynomial-time algorithm. *International Journal of Electrical Power and Energy Systems*, 659–666 (2003)

7. Su, C.-T., Lii, G.-R.: Reliability design of distribution systems using modified Genetic algorithms. *Electric Power System Research*, 201–206 (2002)
8. Su, C.-T., Lii, G.-R.: Reliability planning for composite electric power systems. *Electric Power System Research*, 23–31 (1999)
9. Su, C.-T., Lii, G.-R.: Reliability planning employing genetic algorithms for an electric power system. *Applied Artificial Intelligence*, 763–776 (1999)
10. Ashok Bakkiyaraj, R., Kumarappan, N.: Optimal reliability planning for a composite electric power system based on Monte Carlo simulation using particle swarm optimization. *International Journal of Electrical Power and Energy Systems* 47, 109–116 (2013)
11. Billinton, R., Li, W.: A system state transition sampling method for composite system reliability evaluation. *IEEE Transactions on Power System*, 761–770 (1993)
12. Specht, D.F.: A General Regression Neural Network. *IEEE Transactions on Neural Networks* 2(6), 568–576 (1991)
13. Kim, B., Lee, D.W., Parka, K.Y., Choi, S.R., Choi, S.: Prediction of plasma etching using a randomized generalized regression neural network. *Vacuum* 76, 37–43 (2004)
14. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ, pp. 1942–1948 (1995)
15. El-Abiad, A., Stagg, G.: *Computer methods in power system analysis*. McGraw-Hill (1968)

A Bacteria Foraging-Particle Swarm Optimization Algorithm for QoS Multicast Routing

Rohini Pradhan*, Manas Ranjan Kabat, and Satya Prakash Sahoo

Veer Surendra Sai University of Technology, Burla, Sambalpur, India
{rohinipradhan, sa_tyaparakash}@gmail.com,
manas_kabat@yahoo.com
<http://www.vssut.ac.in>

Abstract. In this paper, we propose a modified Bacteria Foraging-Particles Swarm Optimization (BF-PSO) algorithm for QoS multicast routing. This meta-heuristic optimization algorithm generates a least cost multicast tree under multiple constraints. The algorithm uses random parameters of Particle swarm optimization algorithm to minimize the delay in reaching the global solution. The Simulation results show that the proposed algorithm is more efficient than the existing meta-heuristic algorithms such as Genetic Algorithm, Quantum behaved Particle Swarm Optimization.

Keywords: BFO, multicast tree, PSO, QoS.

1 Introduction

Multicast routing is one of the most important communication techniques over the IP networks for multimedia and real-time applications. These applications include audio/video conferencing, distance learning and televised company meetings. In, multicasting, the source sends a packet only once, even if it needs to be delivered to a large number of receivers, using the network resources optimally. The transfer of information from source to multiple receivers is decided by the Multicast tree. The QoS requirement of the multicast tree is to provide cost sensitive delivery service for the required applications by ensuring sufficient bandwidth, controlling latency and jitter and reducing packet loss.

This QoS multicast routing problem has drawn wide spread attention from researchers who have been using different methods to solve the problem using conventional algorithms, such as exhaustive search routing, greedy routing as well as heuristic and meta-heuristic algorithms. The multicast routing problem is known to be NP-Complete [2].

The Genetic algorithm (GA) is one of the most popular intelligent optimization algorithms applied to QoS Multicast routing problem. Hwang et al. [3] proposed a GA-based algorithm for multicast routing problem. In this algorithm,

* Veer Surendra Sai University of Technology, Burla.

a multicast tree is produced by the means of finding the paths from the source node to each destination node and merging the paths into a tree. The optimization of a multicast tree is achieved through a serial path selection, by crossover and mutation operation. However, when the network is large, the time of searching paths is so long that the algorithm is inefficient. A tree growth based ACO algorithm [4] has been proposed to generate a multicast tree in the way of tree growth and optimizing ant colony parameters through the most efficient combination of various parameters. However, the pheromone updating strategy may lead to converge the solution quickly at the local optima.

Liu et al. [8] and Wang et al. [9] proposed a Particle Swarm Optimization (PSO) based algorithms to solve the QoS Multicast Routing problem by means of serial path selection to realize the optimization of a multicast tree. The multicast tree can obtain a feasible solution by exchanging paths in the vector. A tree based PSO algorithm is proposed by Wang et al. [15] to optimize the multicast tree directly. It is unlike the other conventional methods to finding paths and integrating them to generate a multicast tree. However, the construction of a limited number of multicast trees randomly and simply merging and deleting the loops may not generate a very close to optimal tree. Sun et al. [10] described an algorithm based on the quantum-behaved PSO (QPSO), for QoS multicast routing. The proposed method converts the QoS multicast routing problem into an integer-programming problem and then solves the problem by QPSO. A hybrid PSO with GA operator [17] is used for multicast routing problem. In [17], a set of k -shortest paths are generated as particles and a two point crossover is done on any two randomly chosen particles. It usually happens that this crossover generates a duplicate particle and reduce the searching ability. It again requires an extra effort in removing the duplicate particle and generate a new particle.

The Bacteria Foraging Optimization (BFO) algorithm is one of the new and promising meta-heuristic algorithm proposed by K.M. Passino [18]. The BFO has been used in load dispatch and congestion management problems [22][20]. In this paper, we propose a BF-PSO based algorithm to solve the QoS multicast routing problem. We apply a modified BFO algorithm that uses the parameters of PSO to generate a least cost delay constraint QoS multicast tree. The BFO algorithm selects S number of multicast trees generated as bacterias. Using the process of chemotaxis in which the parameters of PSO algorithm are used to find the tumbling direction, we obtain the optimum multicast tree.

The paper is organised as follows: The problem statement of the QoS multicast routing is presented in the next section. Section 3 presents the proposed BF-PSO based algorithm for QoS multicast routing. The simulation results and analysis is presented in section 4. The section 5, summarises the conclusion of the paper.

2 Problem Statement

The communication network can be represented as an undirected weighted graph $G(V, E)$, where V is a finite set of all nodes that represents routers or switches and E is a finite set of all edges that represents physical or logical connection

between nodes. Let $s \in V$ be the source node of a multicast group, and $M \subseteq V - s$ be a set of destination nodes of the multicast group; such that s and M construct a multicast tree $T(s, M)$. Let R^+ be the set of all real numbers that are nonnegative. The QoS metrics, for each link $e \in E$ can be defined as: Bandwidth function $\implies Bandwidth(e) : E \rightarrow R^+$

Delay function $\implies Delay(e) : E \rightarrow R^+$

Cost function $\implies Cost(e) : E \rightarrow R^+$

Jitter function $\implies Jitter(e) : E \rightarrow R^+$

Loss rate function $\implies LR(e) : E \rightarrow R^+$

Let a multicast tree $T(s, M)$, spans the source node s and the set of destination nodes M , that have following relations. The path from source to any destination $t \in M$ is represented as $P_T(s, t)$:

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

$$Delay(P_T(s, t)) = \sum_{e \in P_T(s, t)} Delay(e)$$

$$Jitter(T(s, M)) = \sqrt{\sum_{e \in T(s, M)} (D(s, t) - delay_avg)^2}$$

Where $delay_avg$ is the average value of delay in the path from the source to the destination nodes.

Loss Ratio of the path is:

$$LR(P_T(s, t)) = 1 - \prod_{e \in P_T(s, t)} (1 - LR(e))$$

$$Bandwidth(P_T(s, t)) = \min\{Bandwidth(e) | e \in P_T(s, t)\}$$

The QoS constraints of the multicast tree is as follows:

1. Delay constraint : $Delay(P_T(s, t)) \leq DC, \forall t \in M$ i.e the maximum delay from source to destination in the multicast tree T should be less than or equal to the delay constraint DC .
2. Jitter constraint : $Jitter(T(s, M)) \leq JC$ i.e delay *Jitter* of the multicast tree should not exceed the jitter constraint JC .
3. Bandwidth constraint : $Bandwidth(P_T(s, t)) \geq BC, \forall t \in M$ i.e the minimum bandwidth of the multicast tree at every link should be greater or equal to the bandwidth constraint BC .
4. Loss ratio constraint : $LR(P_T(s, t)) \leq LRC, \forall t \in M$, i.e the loss rate of the path from source to any destination should be less than or equal to the loss rate constraint LRC .

3 QoS Multicast Routing Based on BFO-PSO

The idea of BFO algorithm is based on the fact that natural selection tends to eliminate animals with poor foraging strategies and favor those having successful foraging strategies [18]. In the process of foraging, *E. coli* bacteria that are present in our intestines undergo four stages, namely, chemotaxis, swarming, reproduction, and elimination and dispersal. In search space, BFO algorithm seeks optimum value through the chemotaxis of bacteria, and realize quorum sensing via assemble function between bacterial, and satisfy the evolution rule of the survival of the fittest make use of reproduction operation, and use elimination-dispersal mechanism to avoiding falling into premature convergence.

An improved Bacteria Foraging Algorithm was proposed by W. Korani in the year 2008, naming it BF-PSO algorithm [19]. Both the algorithms of BFO and PSO are combined in BF-PSO. A unit length direction of tumble characteristic is shown in bacteria is randomly generated in BFO algorithm. This generation of random direction may lead to delay in reaching the global solution. Thus, the PSO ability to exchange the social information and the ability to find a new solution by elimination and dispersal of BFO algorithm is incorporated in BF-PSO. In this algorithm, the global best position and local best position of each bacterium can decide the unit length random direction of tumble behavior of the bacteria. The tumble direction update during the process of chemotaxis loop is determined by:

$$\begin{aligned} \phi(p+1) = & w * \phi(j) + C1 * R1(plbest - pcurrent) \\ & + C2 * R2(pgbest - pcurrent) \end{aligned}$$

where, *pcurrent* is the current position of the bacteria and *plbest* is the local best position of each bacteria and *pgbest* is the global best position of each bacteria.

3.1 Representation and Initialization of Bacteria

In order to generate a QoS multicast tree from the network model, the delay of the links are set to infinite which have the available bandwidth less than the required bandwidth. Then k-least delay paths are generated by using the k-Bellman-Ford algorithm. The delay constrained paths for each destination of the multicast group are selected from the k-least delay paths. For a set of multicast destination nodes, where $M = \{d_1, d_2, \dots, d_m\}$ and a given source node *s*, a bacterium in our proposed BFO algorithm is represented as $b = \{x_1, x_2, \dots, x_m\}$. Each bacterium represents a multicast tree and each bacterium has *m* number of components, i.e. $\{x_1, x_2, \dots, x_m\}$, where the i^{th} component represents the path to the i^{th} destination in a multicast tree. Each component in a bacteria selects a feasible path from k-number of delay constrained paths to each destination.

Each bacterium is a candidate solution of QoS multicast tree from source node to each destination. A bacterium represents a multicast tree after randomly combining the paths from the generated k - least cost delay constrained paths and removing the loops, if any. After the multicast trees are randomly generated and chosen by bacteria, loop deletion and calculation of fitness function is performed.

3.2 A Modified BF-PSO Based Multicast Routing Algorithm

The pseudo code of the modified BF-PSO algorithm is given below:

[Step 1] **Initialization:**

1. S : Number of bacteria in the population.
2. N_c : No. of chemotactic steps.
3. N_s : Swimming length.
4. N_{re} : No. of reproduction steps.
5. N_{re} : No. of reproduction steps.
6. N_{ed} : No. of elimination- dispersal steps.
7. P_{ed} : Probability for elimination-dispersal.
8. $C(i)[i = 1, 2, S]$: Size of step taken in random direction of tumble.
9. $C1, C2, R1, R2, w$: PSO parameters.
10. $P(p, q, r) : \theta^i(p, q, r) | i = 1, 2, ..S$; position of bacteria.
11. Generate random vector $\phi(p)$ whose elements lie between $[1, k]$.

[Step 2] **Elimination-Dispersal loop:** $ell = ell + 1$ from step 3-7.

[Step 3] **Reproduction loop:** $k = k + 1$ step 4-6.

[Step 4] **Chemotaxis loop:** $k = 1.N_s$.

Chemotactic step for every bacterium(i)

Calculation of fitness function: $J(i, j)$ then let $J_{last} = J(i, j)$.

$J_{local}(i, j) = J_{last}(i, j)$

$P(i, j) = P(i, j) + C(i) * \phi(m, i)$

$J(i, j + 1) = Fitness(P(i, j + 1))$

Swim:let $m = 0$

While($n < N_s$)

If $J(i, j + 1) < J_{last}$

$J_{last} = J(i, j + 1)$

$P(i, j + 1) = P(i, j + 1) + C(i) * \phi(m, i)$

$J(i, j + 1) = Fitness(P(i, j + 1))$

$p_{current}(i, j + 1) = P(i, j + 1)$

$J_{local}(i, j + 1) = J_{last}(i, j + 1)$

Else

$p_{current}(i, j + 1) = P(i, j + 1)$

$J_{local}(i, j + 1) = J_{last}(i, j + 1)$

let $m = N_s$

$n = n + 1$

Evaluation of p_{lbest} and p_{gbest} for each bacterium

If ($j \geq 3$) and $p_{gbest}(j) = p_{gbest}(j - 1)$

$$\begin{aligned} &\text{break from chemotactic loop} \\ \phi(m, i + 1) &= w * \phi(m, i) + C1 * R1(plbest - pcurrent) \\ &\quad C2 * R2(pgbest - pcurrent) \end{aligned}$$

[Step 6] **Reproduction:** Compute the health of the bacterium i :

$$j_{health}^i = \min_{j=i}^{N_c+1} \{Fitness(P(i, j))\}$$

[Step 7] **Elimination-Dispersal:** Eliminate and disperse bacteria with probability P_{ed} .

3.3 Bacteria Movement and Calculation of Fitness Function

The movement of bacteria is defined by tumbling and swimming during the process of chemotaxis. In the chemotactic process, the fitness of the i^{th} bacterium, $1 \leq i \leq S$ in the j^{th} chemotactic step, $1 \leq j \leq N_c$ is computed as $Fitness(P(i, j))$. The N_c is the maximum number of chemotactic steps taken during the iteration, where $P(i, j)$ is the current position of the i^{th} bacterium at j^{th} chemotactic step. the bacterium at position $P(i, j)$ represents a multicast tree $T(s, M)$.The fitness function evaluated is stored in the J_{last} . $Fitness(P(i, j))$ is defined as:

$$\begin{aligned} Fitness(T(s, M)) &= Cost(T(s, M)) \\ &\quad + \tau_1 \min(DC - Delay(P_T(s, t)), 0) \\ &\quad + \tau_2 \min(JC - Jitter(P_T(s, t)), 0) \\ &\quad + \tau_3 \min(BC - Bandwidth(P_T(s, t)), 0) \end{aligned}$$

Where τ_1, τ_2, τ_3 are the weights predefined for delay, delay-jitter and bandwidth respectively.

After calculation of the fitness function, we update the local position of the bacterium with respect to the next chemotactic step. During its lifetime, a bacterium must undergo one tumble behavior. The $\phi(m, i)$ decides the direction for the tumble behavior for the bacterium. A bacterium in its chemotactic phase undergoes a maximum of N_s number of swimming steps. During swimming if the updated local position of the bacterium is better than the previous local best position J_{last} , then J_{last} is updated with $J(i, j + 1)$. The PSO parameter $pcurrent$ is evaluated. The current position $pcurrent$ is updated if the current position of the bacterium is better than the previous current position of the bacterium. With each movement the current position of the bacteria is updated as well as the fitness function. Thus each bacterium moves towards the optimum solution of QoS multicast tree in the chemotactic step. The local best position $plocal$ and global best position $pgbest$ of each bacterium is evaluated after the loop swimming

for maximum of consecutive N_s steps. If the global best position $pgbest$ doesn't change over three chemotactic phase then the chemotactic step is stopped and the reproduction phase starts.

Reproduction. In the reproduction phase, the multicast trees obtained in the chemotactic step are sorted and arranged with respect to their fitness values J_{health}^i . The bacterium that has the best fitness value for each chemotactic step are chosen and arranged in ascending order. The bacteria with highest fitness values die and the remaining bacteria with best value remains and copies .

$$J_{health}^i = \min_{j=1}^{N_c+1} \{Func(P(i,j))\}$$

Elimination-Dispersal. In the last step that is in elimination-dispersal step the weakest or the poor bacteria are dispersed with probability of P_{ed} .

4 Simulation Results

In the proposed algorithm, each bacteria represents a possible multicast tree from the source node s to all the destination nodes. If the network is densely connected or the size of the network is large, the number of possible routes of a source-destination pair becomes huge. This makes it obvious that the number of possible routes between two nodes heavily depends on the network topology. An algorithm was designed using the k-shortest path method to automatically generate k number of shortest paths ranging between [5, 20], for each destination. All the experiments were done with Matlab 2009b on Windows 7 and executed on a PC with 2.33 GHz-CPU and 4 GB RAM.

The bandwidth and delay of each link are uniformly distributed in the range [40,80] and [0,30] respectively. The cost of each link is uniformly distributed in the range [5,10]. The proportion of the multicast member nodes is made varied between 10% to 70% for different testing paradigms. The average solution is obtained by running the program 50 times. For comparison, GA-PSO[17], QPSO[16], PSO[9] were tested on the same set of problems.

We used the WAXMAN[1] model in the experiments in order to generate different scale random network topologies. The convergence time of the proposed algorithm is calculated for networks by varying the number of nodes from 20 to 120. The number of destinations are taken as 25% of the number of nodes. The convergence time of the proposed BF-PSO algorithm in comparison to the convergence time of PSO, QPSO and GA-PSO algorithms using the same network topologies is shown in Fig.1. The simulation result shows that the BF-PSO algorithm takes less time in converging than other heuristic algorithms.

We consider a network of 100 nodes to study the performance of the proposed algorithm in terms of multicast tree cost, delay and delay-jitter in comparison to the existing algorithms. The cost, delay and delay-jitter of PSO, QPSO, GA-PSO algorithms were also calculated in the network. The number of destination nodes were taken from 10% to 70%. Fig.2 shows the comparison of multicast tree

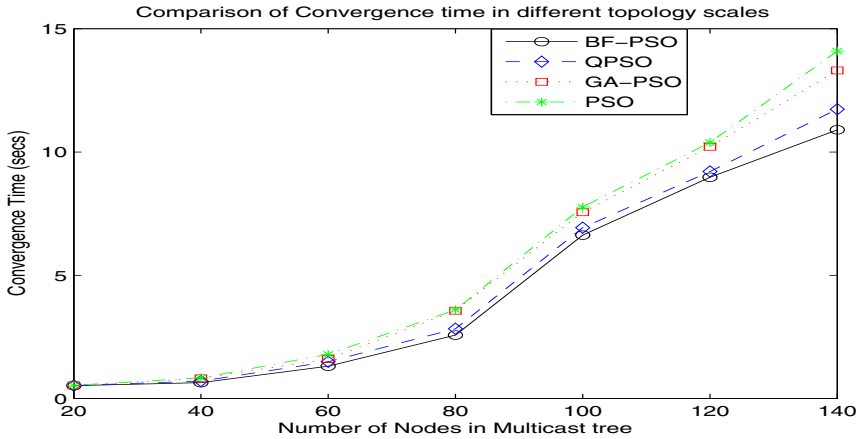


Fig. 1. Convergence time of the multicast tree with 25% destination variable nodes

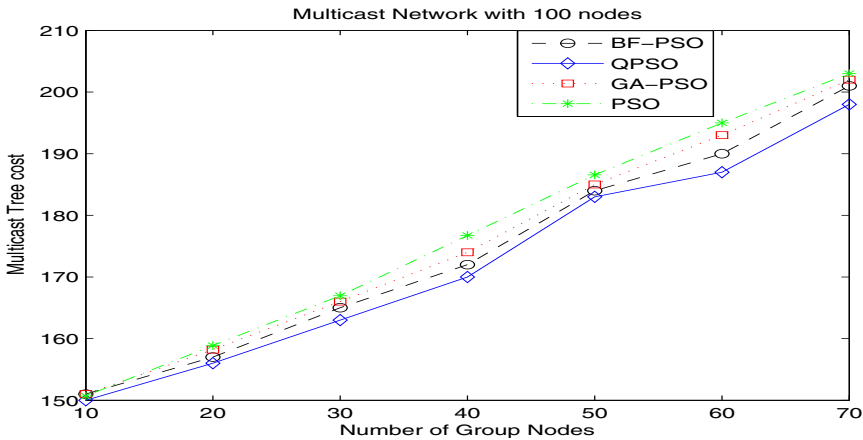


Fig. 2. Cost of the multicast tree

cost of the proposed algorithm with the existing algorithms. The result reflects that the cost of PSO is highest while QPSO has the lowest cost. BF-PSO has a better cost than PSO and GA-PSO. Fig.3 shows the delay of the multicast generated by each algorithm that is averaged over different destination nodes with 100 network nodes. It is observed that the delay of the multicast tree generated by our proposed algorithm is less than the existing algorithms. Fig.4 shows the delay-jitter of the multicast tree generated by each algorithm in a 100 network nodes. It is also observed that the delay-jitter is less than all the other algorithms.

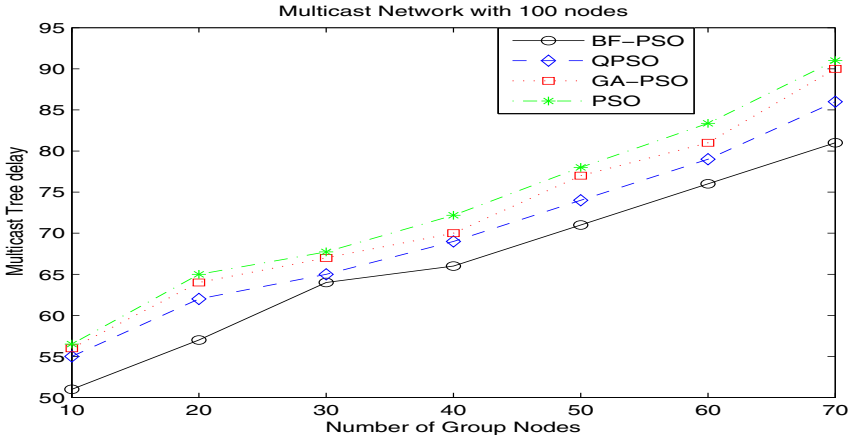


Fig. 3. Delay of the multicast tree with 100 nodes

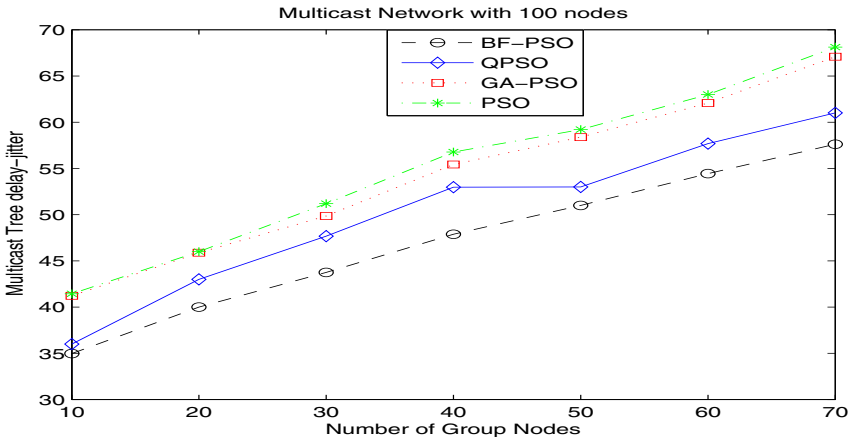


Fig. 4. Delay-jitter of the multicast tree with 100 nodes

It can be thus observed that the multicast tree obtained by the BF-PSO has cost more than the QPSO but comparatively less delay and delay-jitter showing that BF-PSO can generate the higher quality solutions on the multicast routing problems.

5 Conclusions

In this paper, a novel multicast routing algorithm based BF-PSO algorithm using random velocity parameters of PSO is proposed. The algorithm utilizes the discrete PSO algorithm to optimally search the solution space for the optimal multicast tree satisfying the quality of service requirement. To verify the performance of the proposed algorithms, simulations were carried out with different

sizes of multicast groups on diverse topology networks. Experimental results show that the algorithm is feasible and effective. It not only converges fast, but also can escape from local optimum and effectively search for the global optimum.

References

1. Waxman, B.M.: Routing of multipoint connections. *IEEE J. Select Areas Commun.* 6(9), 1617–1622 (1988)
2. Wang, Z., Crowcroft, J.: Quality of service for supporting multi-media application. *IEEE J. Select Areas Commun.* 14, 1228–1234 (1996)
3. Ren, K., Zeng, K., Lou, W.: Multicast routing based on genetic algorithms. *J. Inform. Sci. Eng.*, 885–901 (2000)
4. Wang, H., Shi, Z., Ma, J., et al.: The tree-based ant colony algorithm for multi-constraints multicast routing. In: *Proceedings of the 9th International Conference on Advanced Communication Technology (ICACT 2007)*, February 12–14, vol. 3, pp. 1544–1547. IEEE, Seoul (2007)
5. Wang, H., et al.: A tree-growth based ant colony algorithm for QoS multicast routing problem. *Expert Systems with Applications* (2011), doi:10.1016/j.eswa.03.065
6. Wang, Y., Xie, J.Y.: Ant colony optimization for multicast routing. In: *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS 2000)*, December 4–6, pp. 54–57. IEEE, Tianjin (2000)
7. Chu, C.H., Gu, J.H., Hou, X.D., et al.: A heuristic ant algorithm for solving QoS multicast routing problem. In: *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)*, May 12–17, vol. 2, pp. 1630–1635. IEEE, Honolulu (2002)
8. Liu, J., Sun, J., Xu, W.-b.: QoS multicast routing based on particle swarm optimization. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) *IDEAL 2006*. LNCS, vol. 4224, pp. 936–943. Springer, Heidelberg (2006)
9. Wang, Z., Sun, X., Zhang, D.: A PSO-based multicast routing algorithm. In: *Proceedings of Third International Conference on Natural Computation (ICNC)*, pp. 664–667 (2007)
10. Sun, J., Liu, J., Xu, W.-b.: QPSO-based QoS multicast routing algorithm. In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H.A., Iba, H., Chen, G.-L., Yao, X. (eds.) *SEAL 2006*. LNCS, vol. 4247, pp. 261–268. Springer, Heidelberg (2006)
11. Panigrahi, B.K., Ravikumar Pandi, 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)
12. Panigrahi, B.K., Ravi Kumar Pandi, V.: An Improved Adaptive Particle Swarm Optimization Approach for Multi Modal Function Optimization. *International Journal of Information & Optimization Sciences* 29(2), 359–375 (2008)
13. Xi-Hong, C., Shao-Wei, L., Jiao, G., Qiang, L.: Study on QoS multicast routing based on ACO-PSO algorithm. In: *Proceedings of 2010 International Conference on Intelligent Computation Technology and Automation*, pp. 534–537 (2010)
14. Ghosh, S., Das, S., Kundu, D., Suresh, K., Panigrahi, B.K., Cui, Z.: An inertia-adaptive particle swarm system with particle mobility factor for improved global optimization. *Neural Computing and Applications* 21(2), 237–250 (2012)

15. Wang, H., Meng, X., Li, S., Xu, H.: A tree-based particle swarm optimization for multicast routing. *Computer Network* 54, 2775–2786 (2010)
16. Sun, J., Fang, W., Wu, X., Xie, Z., Xu, W.: QoS multicast routing using a quantum-behaved particle swarm optimization algorithm. *Engineering Applications of Artificial Intelligence* 24, 123–131 (2011)
17. Abdel-Kader, R.F.: Hybrid discrete PSO with GA operators for efficient QoS-multicast routing. *International Journal of Hybrid Information Technology* 4(2) (April 2011)
18. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Sys. Mag.* 22(3), 52–67 (2002)
19. Mishra, S., Tripathy, M.: Bacteria Foraging-Based Solution to Optimize Both Real Power Loss and Voltage Stability Limit. *IEEE Transactions on Power Systems* 22(1) (February 2007)
20. Panigrahi, B.K., Ravikumar Pandi, V.: A Bacterial Foraging Optimization Nelder Mead Hybrid Algorithm for Economic Load Dispatch. *IET Proceedings of Gen. Trans. and Distribution* 2(4), 556–565 (2008)
21. Korani, W.: Bacterial Foraging Oriented by Partical Swarm Optimization Strategy for PID Tuning. In: *GECCO 2008, Atlanta, Georgia, USA, July 12-16 (2008)*
22. Panigrahi, B.K., Ravikumar Pandi, V.: Congestion management using adaptive bacterial foraging algorithm. *International Journal on Energy Conversion and Management* 50, 1202–1209 (2009)

Performance Evaluation of Particle Swarm Optimization Algorithm for Optimal Design of Belt Pulley System

Pandurengan Sabarinath¹, M.R. Thansekhar¹, and R. Saravanan²

¹ K.L.N College of Engineering, Pottapalayam– 630611, India
 jananisabari@gmail.com, thansekhar@yahoo.com

² JCT College of Engineering and Technology, Coimbatore, India
 saradharani@hotmail.com

Abstract. The present scenario in the design of machine elements includes the minimization of weight of the individual components in order to reduce the overall weight of the machine elements. It saves both cost and energy involved. Belts are used to transmit power from one shaft to another by means of pulleys which rotate at the same speed or different speeds. Generally, the weight of pulley acts on the shaft and bearings. In the present study, minimization of weight of a belt pulley system has been investigated. Particle swarm optimization algorithm (PSO) is used to solve the above mentioned problem subjected to a set of practical constraints and it is compared with the results obtained by Differential Evolution Algorithm (DEA). Our results indicate that PSO approach handles our problem efficiently in terms of precision and convergence and it outperforms the results presented in the literature.

Keywords: Optimal Design, Belt pulley system, Particle swarm optimization algorithm.

| Nomenclature | |
|--|---|
| b : Width of the pulley, cm | pbest : particle's best |
| c ₁ : Cognitive parameter | rand1 : Random number between 0 and 1 |
| c ₂ : Social parameter | rand2 : Random number between 0 and 1 |
| CR : Crossover Constant | t _b : thickness of the belt, cm |
| d _p : dia of any pulley, cm | t ₁ : thickness of first pulley, cm |
| d ₁ : dia of first pulley, cm | t ₁ ¹ : thickness of third pulley, cm |
| d ₁ ¹ : dia of third pulley, cm | t ₂ : thickness of second pulley, cm |
| d ₂ : dia of second pulley, cm | t ₂ ¹ : thickness of fourth pulley, cm |
| d ₂ ¹ : dia of fourth pulley, cm | T ₁ : tension at the tight side, kgf |
| g _{best} : global best | T ₂ : tension at the slack side, kgf |
| N ₁ : rpm of first pulley | V : tangential velocity of pulley, cm/s |
| N ₁ ¹ : rpm of third pulley | W _p : Weight of pulleys, kg |
| N ₂ : rpm of second pulley | w : inertia weight |
| N ₂ ¹ : rpm of fourth pulley | X _j ^{min} : Lower bound of j th decision parameter |
| N _p : rpm of any pulley | X _j ^{max} : Upper bound of j th decision parameter |
| NP : Population Size | ρ : density of the material, kg/cm ³ |
| P : power transmitted by belt pulley drive, hp | σ _b : allowable tensile stress of belt material, kg/cm ² |

1 Introduction

The most important problem that confronts practical engineers is the mechanical design, a field of creativity. Mechanical design can be defined as the selection of materials and geometry, which satisfies the specified and implied functional requirements while remaining within the confines of inherently unavoidable limitations. Engineering design can be characterized as a goal-oriented, constrained, decision making process to create products that satisfy well-defined human needs. Design optimization consists of certain goals (objective functions), a search space (feasible solutions) and a search process (optimization methods). The feasible solutions are the set of all designs characterized by all possible values of the design parameters (design variables). The optimization method searches for the optimal design from all available feasible designs. Mechanical design includes an optimization process in which designers always consider certain objectives such as strength, deflection, weight, wear, corrosion, etc depending on the requirements. However, design optimization for a complete mechanical assembly leads to a complicated objective function with a large number of design variables. So it is good practice to apply optimization techniques for individual components or intermediate assemblies rather than a complete assembly. For example, in an automobile power transmission system, the optimization of the gearbox is computationally and mathematically simpler than the optimization of the complete transmission system. Analytical or numerical methods for calculating the extremes of a function have long been applied to engineering computations. Although these methods perform well in many practical cases, they may fail in more complex design situations. In real design problems the number of design variables can be very large, and their influence on the objective function to be optimized can be very complicated, with a nonlinear character. The objective function may have many local optima, whereas the designer is interested in the global optimum. Such problems cannot be handled by classical methods (e.g. gradient methods) that only compute local optima. So there remains a need for efficient and effective optimization methods for mechanical design problems. Continuous research is being conducted in this field and nature-inspired heuristic optimization methods are proving to be better than deterministic methods and thus are widely used.

Design optimization can be defined as the process of finding the maximum or minimum of some parameter, which may be called the objective function and it must also satisfy a certain set of specified requirements called constraints. Many methods have been developed and are in use for design optimization. All of these methods use mathematical programming. A belt is a loop of flexible material used to mechanically link two or more rotating shafts. Belts may be used as a source of motion, to transmit power efficiently, or to track relative movement. Belts are looped over pulleys. In a two pulley system, the belt can either drive the pulleys in the same direction, or the belt may be crossed, so that the direction of the shafts is opposite. As a source of motion, a conveyor belt is one application where the belt is adapted to continuously carry a load between two points. A pulley is a wheel on an axle that is designed to support movement of a cable or belt along its circumference. Pulleys are used in a variety of ways to lift loads, apply forces, and to transmit power. A belt and pulley system is characterized by two or more pulleys in common to a belt. This allows for

mechanical power, torque and speed to be transmitted across axles. If the pulleys are of differing diameters, a mechanical advantage is realized.

There are many nature-inspired optimization algorithms, such as the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Harmony Search (HS), the Grenade Explosion Method (GEM), etc., working on the principles of different natural phenomena. Particle swarm optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart in 1995. The underlying motivation for the development of PSO algorithm was social behavior of animals such as bird flocking, fish schooling, and swarm theory. Similar to genetic algorithms (GA), PSO is a population based optimization tool, both have fitness values to evaluate the population, both update the population and search for the optimum with random techniques, both systems do not guarantee success. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, particles update themselves with internal velocity. They also have memory, which is important to the algorithm. Also, the potential solutions, called particles, are “flown” through the problem space by following the current optimum particles. Compared to GA, the information sharing mechanism in PSO is significantly different. In GAs chromosomes share information with each other. So the whole population moves like a group toward an optimal area. In PSO, only Gbest gives out the information to others. It is a one-way information sharing mechanism. The evolution only looks for the best solution. Compared with GA, all the particles tend to converge to the best solution quickly even in the local version in most cases. The advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas, such as function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied.

The problem is formulated with single objective of minimizing weight of a belt pulley system without compromising specified strength. The primary optimization variables are: the diameters of the driver and driven pulley and width of the pulley. In this paper, PSO is employed in continuous optimization of engineering design of a belt pulley system subjected to a set of practical constraints. The constraints considered in this problem are power to be transmitted and size constraints. Formulation of fitness function as applicable to the present study is also presented. PSO is fairly a new non-traditional optimization technique and the algorithm is explained in detail. However, a brief description is included for the sake of easiness.

2 Literature Review

The importance and application of various traditional optimization techniques to solve many real world design optimization problems taken from wide range of industries have been discussed in [1]. The procedure and the necessary steps to design various mechanical elements and transmission elements with design calculations are dealt in [2]. The theory and applications of traditional optimization techniques such as linear and nonlinear programming, dynamic programming, integer programming and stochastic programming along with recently developed techniques such as Genetic Algorithm, Simulated annealing and neural network based fuzzy optimization techniques have been discussed with solved examples in [3]. Formulation of the

optimization problem for finding the width and diameters of the step of a four step cone pulley for minimum weight is discussed in [3]. The objective function of minimizing the weight of the belt pulley system is taken from [4] in which Reddy used geometric programming method to solve the optimal design of belt pulley system. The same problem is solved by using a non-traditional optimization technique namely Genetic Algorithm (GA) for getting the solution in [5]. The authors have also solved volume minimization of helical compression spring and weight minimization of hollow shaft using GA. GA uses the theory of Darwin based on the survival of the fittest. GA provides a near optimal solution for a complex problem having large number of variables and constraints. This is mainly due to the difficulty in determining the optimum controlling parameters such as population size, crossover rate and mutation rate [6]. ABC uses the foraging behavior of a honey bee [7]. Inspired by the bee behavior, Artificial Bee Colony [8] is one of the generally applicable techniques used for optimizing numerical functions and real-world problems. Compared with GA and other similar evolutionary techniques, ABC has some attractive characteristics and in many cases proved to be more effective [8]. ACO works on the behavior of an ant in searching for a destination from the source [9]. This co-operative search behavior of real ants inspired the new computational paradigm for optimizing real life systems and it is suited to solving large-scale optimization problems. ACO has also been applied to other optimization problem like the travelling salesman problem and scheduling [10]. Harmony Search works on the principle of music improvisation in music players [11] and GEM works on the principle of the explosion of a grenade [12]. These algorithms have been applied to many engineering optimization problems and proved effective in solving some specific kinds of problem.

If individual particles in a PSO system have quantum behavior, the PSO algorithm is bound to work in a different fashion [14]. Similarly to Genetic Algorithm, Particle swarm optimization (PSO) is an optimization tool based on a population, where each member is seen as a particle, and each particle is a potential solution to the problem under analysis [15]. Jerald et al designed different scheduling mechanisms to generate optimum scheduling using non-traditional approaches such as genetic algorithm (GA), simulated annealing (SA) algorithm, memetic algorithm (MA) and particle swarm algorithm (PSA) by considering multiple objectives, i.e., minimizing the idle time of the machine and minimizing the total penalty cost for not meeting the deadline concurrently. Noorul Haq et al [17] used particle swarm optimization (PSO) for the optimal machining tolerance allocation of over running clutch assembly to obtain the global optimal solution. The objective was to obtain optimum tolerances of the individual components for the minimum cost of manufacturing. The result obtained by PSO was compared with the geometric programming (GP) and genetic algorithm (GA) and the performance of the result was analyzed. Saravanan et al [18] solved the problem of optimal machining parameters for continuous profile machining with respect to the minimum production cost subject to a set of practical constraints. Due to high complexity of this machining optimization problem, six non-traditional algorithms, the genetic algorithm (GA), simulated annealing algorithm (SA), Tabu search algorithm (TS), memetic algorithm (MA), ants colony algorithm (ACO) and the particle swarm optimization (PSO) were employed to resolve the problem. The results obtained from GA, SA, TS, ACO, MA and PSO were compared for various profiles.

For solving constrained optimization problems, many constraint handling methods such as static penalty, dynamic penalty, Adaptive penalty function etc has been proposed. In this work, Deb's constraint handling rule [19] is used for solving the problem. The detailed procedure for the implementation of Differential Evolutionary Algorithm is presented in [20]. Gnanambal et al [21] used hybridization of DE and PSO (DEPSO) for determining the maximum loadability limit of power system.

In this research paper, a particle swarm optimization algorithm is used for solving the weight minimization of belt pulley system problem attempted by the authors [5] and the results are found to be closer to the global optimum.

3 Mathematical Formulation

3.1 Optimum Design of Belt Pulley System

The belts are used to transmit power from one shaft to another by means of pulleys which rotate at the same speed or different speeds. The stepped flat belt drives are mostly used in factories and workshops where the moderate amount of power is to be transmitted. Generally, the weight of pulley acts on the shaft and bearings. The shaft failure is most common due to the weight of pulleys. In order to prevent the shaft and bearing failure, weight minimization of the flat belt drive is essential. The schematic representation of the belt pulley system is presented in Fig.1

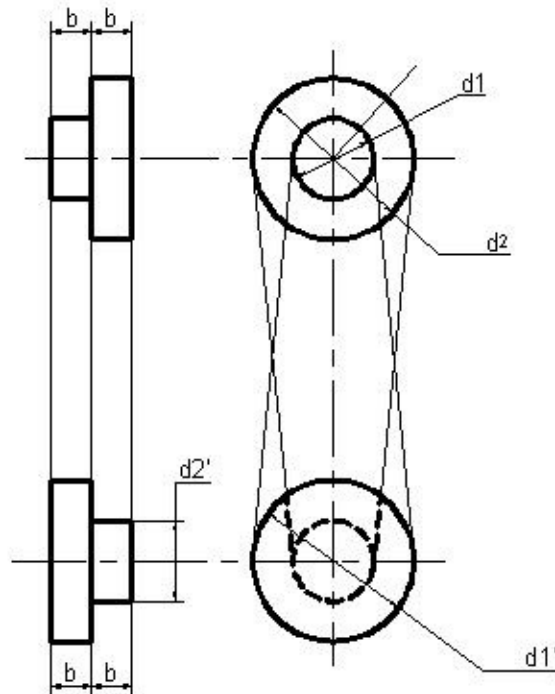


Fig. 1. Schematic Representation of Belt Pulley System

3.2 Objective Function

The mathematical model proposed by [5] is used in this work. The weight of the pulley is considered as objective function which is to be minimized as

$$W_p = \pi \rho b [d_1 t_1 + d_2 t_2 + d_1^1 t_1^1 + d_2^1 t_2^1] \quad (1)$$

Assume the thickness of pulley as 10% of its diameter. Hence $t_1=0.1d_1$, $t_2=0.1d_2$, $t_1^1=0.1d_1^1$, $t_2^1=0.1d_2^1$

According to Velocity ratio of belt drive

$$\frac{d_1}{d_1^1} = \frac{N_1^1}{N_1} \quad (2)$$

Substituting the values of $N_1 = 1000$ rpm and $N_1^1 = 500$ rpm in the above equation we get $d_1^1 = 2 d_1$

Similarly

$$\frac{d_2}{d_2^1} = \frac{N_2^1}{N_2} \quad (3)$$

Substituting the values of $N_2 = 250$ rpm and $N_2^1 = 500$ rpm in the above equation we get $d_2^1 = 0.5 d_2$

Substituting the above relationships in equation (1), we get

$$W_p = \pi \rho b \times 0.1 \left[d_1^2 + d_2^2 + (d_1^1)^2 + (d_2^1)^2 \right]$$

$$W_p = \pi \rho b \times 0.1 \left[d_1^2 + d_2^2 + (2d_1)^2 + (0.5d_2)^2 \right]$$

By taking $\rho = 7.2 \times 10^{-3}$ Kg/cm³ and simplifying the above equation, the objective function can be written as

$$W_p = [0.011309733d_1^2 + 0.0028274d_2^2]b \quad (4)$$

It is subjected to the following constraints:

The transmitted power (P) can be represented as

$$P = \frac{(T_1 - T_2)}{75} \cdot V \quad (5)$$

Substituting the expression for V in the above equation, one gets

$$P = (T_1 - T_2) \frac{\pi d_p N_p}{75 \times 60 \times 100} \quad (6)$$

$$P = T_1 \left(1 - \frac{T_2}{T_1} \right) \frac{\pi d_p N_p}{75 \times 60 \times 100} \quad (7)$$

Assuming $T_2 / T_1 = 0.5$, $P = 10$ hp and substituting the above values, one gets

$$10 = T_1 (1 - 0.5) \frac{\pi d_p N_p}{75 \times 60 \times 100} \quad (8)$$

$$T_1 = \frac{2864789}{d_p N_p} \quad (9)$$

Assuming $d_2 N_2 < d_1 N_1$ and

$$T_1 < \sigma_b b t_b \quad (10)$$

$$\sigma_b b t_b \geq \frac{2864789}{d_2 N_2} \quad (11)$$

Substituting $\sigma_b = 30 \frac{kg}{cm^2}$, $t_b = 1$ cm, $N_2 = 250$ rpm in the above equation, one gets

$$b \geq \frac{381.97}{d_2} \quad \text{or} \\ b d_2 - 381.97 \geq 0 \quad (12)$$

Assuming the width of the pulley either less than or equal to one fourth of the dia of first pulley, the constraint is expressed as

$$b \leq 0.25 d_1 \quad \text{or} \\ \frac{d_1}{4b} - 1 \geq 0 \quad (13)$$

The ranges of variables are mentioned as follows

$$15 \leq d_1 \leq 25, \\ 70 \leq d_2 \leq 80, \\ 4 \leq b \leq 10$$

4 Particle Swarm Optimization

4.1 Introduction

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995, inspired by the social behavior of bird flocking or fish schooling [13]. The particle swarm concept originated as a simulation of a simplified social system. The original intent was to graphically simulate the choreography of a bird of a bird flock or fish school. However, it was found that the particle swarm model can be used as an optimizer. As stated before, PSO simulates the behaviors of bird flocking. Suppose the following scenario: a group of birds are randomly searching for food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. However, they know how far the food is in each iteration. So what's the best

strategy to find the food? The effective one is to follow the bird which is nearest to the food. PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a “bird” in the search space. We call it a “particle”. All of the particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two “best” values. The first one is the best solution (fitness) it has achieved so far. The fitness value is stored. This value is called “pbest”.

Another “best” value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the population. This best value is a global best and is called “gbest”. After finding the two best values, the particle updates its velocity and positions. Eberhart and Shi [15] have introduced an inertia weight factor that dynamically adjust the velocity over time, gradually focusing the PSO into a local search, the particle updates its velocity and positions with the following equation:

$$v[] = w*v[] + c_1 * rand() *(pbest[] - present[]) + c_2 * rand() *(gbest[] - present[]) \quad (14)$$

$$present [] = present [] + v[] \quad (15)$$

where

$v[]$ is the particle velocity,

$present[]$ is the current particle (solution),

$pbest[]$ is the Particle’s best,

$gbest[]$ is the global best,

$rand()$ is a random number between (0,1), and c_1, c_2 are learning factors, and usually, $c_1 = c_2 = 2$.

4.2 PSO Parameter Control

In PSO, the following are the parameters that need to be tuned. Here is a list of the parameters and their typical values used in this work based on the model reported in [18]:

a) The number of particles:

The typical range is 20–40. Actually for most of the problems, 10 particles are large enough to get good results. For some difficult or special problems, one can try 100 or 200 particles as well.

b) Dimension of particles:

Particles dimension is determined by the problem to be optimized. Here, diameter of the driver pulley (d_1), diameter of the driven pulley (d_2) and the width of the pulley (b) are taken as the dimension of particles.

c) Range of particles:

This is also determined by the problem to be optimized.

d) Learning factors:

c_1 and c_2 are usually equal to 2. However, other settings are also used in different papers. But usually c_1 equals to c_2 and ranges from [0, 4]

e) The stop condition:

The maximum number of iterations the PSO executes. This stop condition depends on the problem to be optimized.

f) *Inertia weight:*

This is the weight given to the previous velocity. Inertia weight w is based on a gradual decreasing from 0.9 to 0.4 with a linear decreasing rate

4.3 PSO Algorithm

PSO implements the foraging behavior of a bird for searching food. The swarm intelligence is an emerging research field that presents features of self-organization and cooperation principles among group members bio-inspired on social insect societies [13]. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two “best” values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called p_{best} . Another “best” value that is tracked by the particle swarm optimizer is the best value, which is the best value obtained so far by any particle in the population. This best value is a global best and called g_{best} . The equations (14) & (15) are used to update the velocity and position of the particle. The belt pulley system optimization model using PSO algorithm is shown as a flowchart in Fig.2

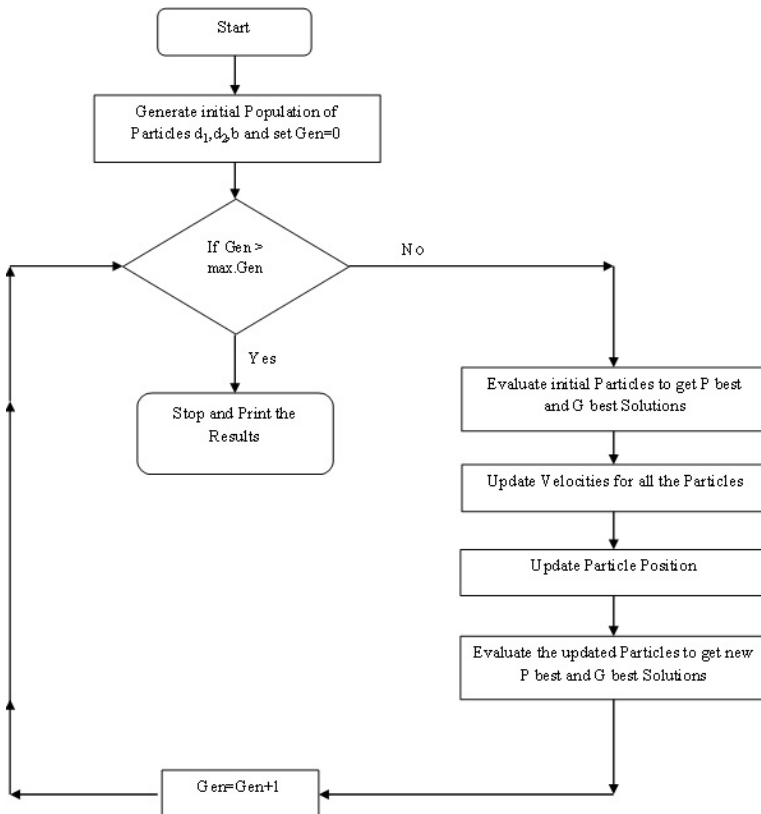


Fig. 2. Flow chart of PSO Algorithm

4.4 Constraint Handling

It is seen from the above steps that no provision is made to handle the constraints in the problem. Many types of constraint handling technique are available in the literature, such as incorporation of static penalties, dynamic penalties, adaptive penalties etc. Deb's heuristic constrained handling method [19] is used in the proposed PSO method. This method uses a tournament selection operator in which two solutions are selected and compared with each other. The following three heuristic rules are implemented on them for the selection:

- If one solution is feasible and the other infeasible, then the feasible solution is preferred.
- If both the solutions are feasible, then the solution having the better objective function value is preferred.
- If both the solutions are infeasible, then the solution having the least constraint violation is preferred.

These rules are implemented at the end of the updating of particle's position and velocity.

5 Differential Evolution Algorithm

Differential evolution, a stochastic, simple yet powerful evolutionary algorithm, not merely possesses the advantage of a quite few control variables but also performs well in convergence was introduced to solve the global optimization by Storn and Price [20]. DE creates new candidate solutions by perturbing the parent individual with the weighted difference of several other randomly chosen individuals of the same population. A candidate replaces the parent only if it is better than its parent. Thereafter, DE guides the population towards the vicinity of the global optimum through repeated cycles of mutation, crossover and selection. The main procedure of DE [21] is explained in detail as follows.

5.1 Initialization

This is the first step in DE. Typically, each decision parameter in every vector of the initial population is assigned a randomly chosen value from within its corresponding feasible bounds.

$$x_{j,i}^{(G=0)} = x_j^{\min} + \text{rand}_j [0,1] (x_j^{\max} - x_j^{\min})$$

where $i = 1, \dots, NP$ and $j = 1, \dots, D$. $x_{j,i}^{(G=0)}$ is the initial value ($G = 0$) of the j^{th} parameter of the i^{th} individual vector. Once every vector of the population has been initialized, its corresponding fitness value is calculated and stored for future reference.

5.2 Mutation

The DEA optimization process is carried out by applying the following three basic genetic operations; mutation, recombination (also known as crossover) and selection. After the population is initialized, the operators of mutation, crossover and selection create the population of the next generation $P^{(G+1)}$ by using the current population $P^{(G)}$. At every generation G , each vector in the population has to serve once as a target vector $X_i^{(G)}$, the parameter vector has index i , and is compared with a mutant vector. The mutation operator generates mutant vectors ($V_i^{(G)}$) by perturbing a randomly selected vector (X_{r1}) with the difference of two other randomly selected vectors (X_{r2} and X_{r3}).

$$v_i^G = x_{r1}^G + F \cdot (x_{r2}^G - x_{r3}^G);$$

$$i = 1, \dots, NP$$

Vector indices $r1$, $r2$ and $r3$ are randomly chosen, which $r1$, $r2$ and $r3 \in \{1, \dots, NP\}$ and $r1 \neq r2 \neq r3 \neq i$. X_{r1} , X_{r2} and X_{r3} are selected anew for each parent vector. F is scaling mutation factor.

5.3 Crossover

In this step, crossover operation is applied in DEA because it helps to increase the diversity among the mutant parameter vectors. At the generation G , the crossover operation creates trial vectors (U_i) by mixing the parameters of the mutant vectors (V_i) with the target vectors (X_i) according to a selected probability distribution.

$$U_i^{(G)} = u_{j,i}^{(G)} = \begin{cases} v_{j,i}^{(G)} & \text{if } \text{rand}_j(0,1) \leq CR \text{ or } j=s \\ X_{j,i}^{(G)} & \text{otherwise} \end{cases}$$

The crossover constant CR is usually selected from within the range $[0, 1]$. The crossover constant controls the diversity of the population and aids the algorithm to escape from local optima. rand_j is a uniformly distributed random number within the range $(0,1)$ generated anew for each value of j . s is the trial parameter with randomly chosen index $\varepsilon \in \{1, \dots, D\}$, which ensures that the trial vector gets at least one parameter from the mutant vector.

5.3.1 Selection

The selection operator chooses the vectors that are going to compose the population in the next generation. This operator compares the fitness of the trial vector and the corresponding target vector and selects the one that provides the best solution. The fitter of the two vectors is then allowed to enter into the next generation.

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

The DEA optimization process is repeated across generations to improve the fitness of individuals. The belt pulley system optimization model using DE algorithm is shown as a flowchart in Fig.3

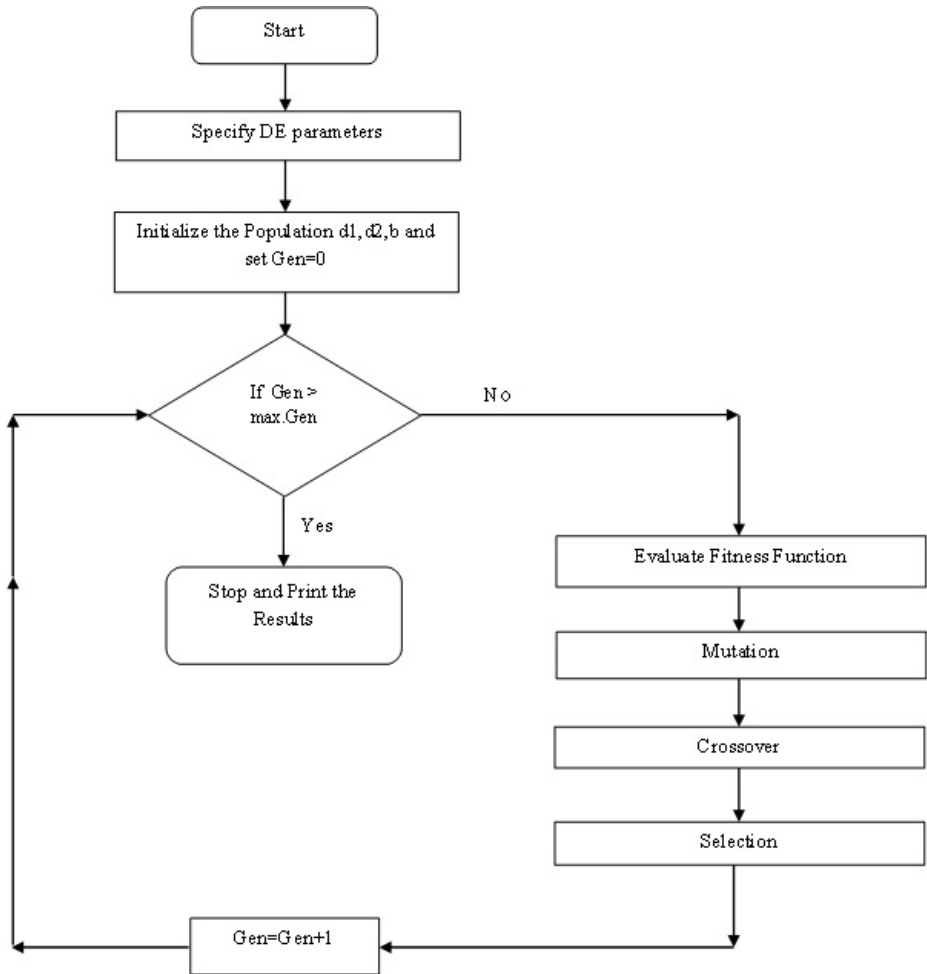


Fig. 3. Flow chart of DE Algorithm

6 Results and Discussion

To study the performance of PSO approach, constrained optimization of input parameters to minimize the weight of belt pulley system was solved and the best results obtained through the mentioned optimization approaches in 20 trials were compared with those reported in the literature. PSO method was implemented using

JAVA to run on a PC compatible with Pentium IV, a 3.2 GHz processor and 2 GB of RAM (Random Access Memory). In the experiment, to start PSO approach, the population size is set to 20 particles, and the maximum generations (set as stopping condition) is set to 200 generations. A total of 4000 fitness function evaluations were made with this optimization approach in each run. The program was executed 20 times to see the convergence characteristics of PSO algorithm. In this work, inertia weight w is based on a gradual decreasing from 0.9 to 0.4 with a linear decreasing rate. Better results can be obtained as per literature, if we gradually decrease the value of w with a linear decreasing rate. The convergence of the results obtained by PSO algorithm is shown in Fig.4.

In order to compare the performance of PSO algorithm, the same problem was solved by DE algorithm. Both PSO and DE algorithms were implemented using Matlab 7 to run on a PC compatible with Pentium IV, a 3.2 GHz processor and 2 GB of RAM (Random Access Memory). In this experiment, to start DE approach, the population size is set to 50, and the maximum generations (set as stopping condition) is set to 200 generations. The crossover constant is set to 0.8 and scaling mutation factor is set to 1. Static penalty method is applied for handling the constraints. A total of 10000 fitness function evaluations were made with this optimization approach in each run. The program was executed 20 times to see the convergence characteristics of DE algorithm. The convergence of the results obtained by DE algorithm is shown in Fig.5. From figure 5, it is evident that the result converges to 104.76 kg after 100 generations and the best value obtained is better than the published result of 105.2 kg and higher than the result reported for GA and PSO.

The results obtained by the implementation of both PSO and DE algorithms have been compared with the results obtained by GA and published result in Table 1. From Table 1, it is evident that the best value of 100.6733 kg reported by PSO algorithm is better than the results obtained by DEA, GA and published results. The statistical performance of both PSO and DE algorithms implemented in this work has been presented in Table 2. The number of fitness function evaluations is less for PSO algorithm than DEA. It clearly indicates that PSO algorithm performs well for the belt pulley system with minimum number of function evaluations than DEA.

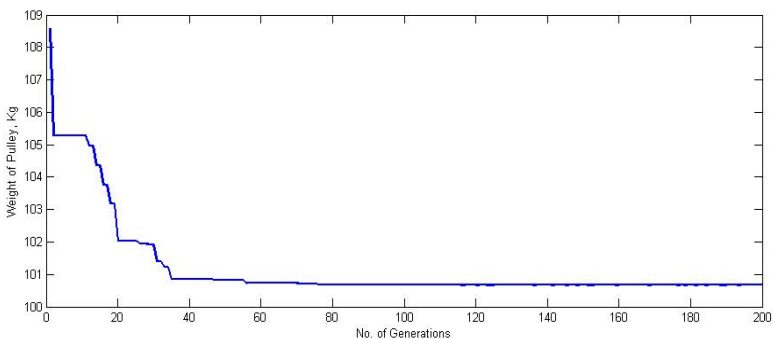


Fig. 4. Evolution of best mean results obtained by PSO algorithm

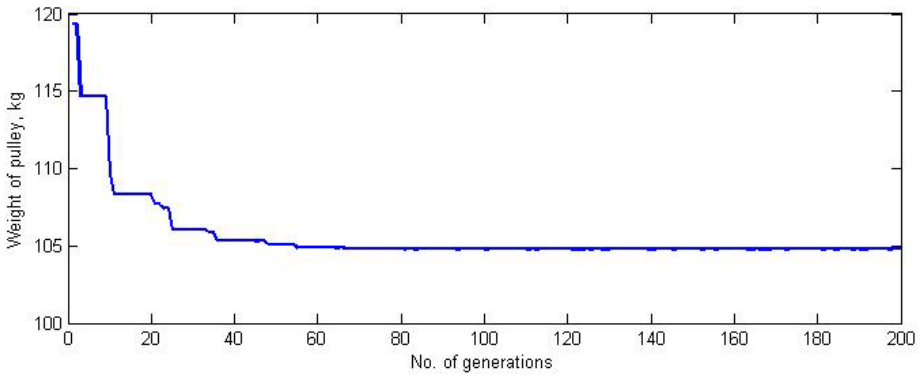


Fig. 5. Evolution of best mean results obtained by DE algorithm

Table 1. Comparison of Results obtained by PSO with DE, GA and the Published results

| Optimal Values | Results obtained by PSO | Results obtained by DE | Results obtained by GA | Published Result |
|---------------------------|-------------------------|------------------------|------------------------|------------------|
| Pulley dia (d_1), cm | 20.7217 | 21.0014 | 20.957056 | 21.12 |
| Pulley dia (d_2), cm | 71.8031 | 72.7514 | 72.906562 | 73.25 |
| Pulley dia (d_1'), cm | 41.4434 | 42.0028 | 42.370429 | 42.25 |
| Pulley dia (d_2'), cm | 35.9015 | 36.3757 | 36.453281 | 36.60 |
| Pulley width (b), cm | 5.1804 | 5.2503 | 5.239177 | 5.21 |
| Pulley weight, Kg | 100.6733 | 104.7602 | 104.533508 | 105.2 |

Table 2. Statistical performance of belt pulley system

| Method | Best | Worst | Mean | Standard Deviation | Evaluations |
|--------|----------|----------|----------|--------------------|-------------|
| PSO | 100.6733 | 106.8759 | 101.8524 | 0.4912 | 80000 |
| DE | 104.7602 | 118.4532 | 105.2154 | 0.6999 | 200000 |

7 Conclusion

In this paper, PSO approach is proposed and applied to solve engineering design problem i.e., constrained optimization of input parameters to minimize the weight of belt pulley system. In order to compare the performance of PSO algorithm for the belt pulley problem, Differential Evolutionary Algorithm is applied to solve the same problem. The simulation results presented in this paper demonstrate that PSO approach tested is an efficient method to improve the PSO's performance in preventing premature convergence to local minima. The proposed PSO approach performed consistently well in our work, with better results than the results obtained

by DE, GA and previously published solutions for these problem. In terms of convergence, the simulation results show that the PSO converge to obtain solutions closer to the good solution and present a small standard deviation. Future work will consider improved PSO variants (CLPSO, FIPS, and OLPSO) and other methods such as ABC, HS, TLBO etc. Furthermore, simulated annealing local search can be combined with PSO as hybrid technique for constrained problems so that better solutions can be obtained as a future extension of the present work.

References

1. Deb, K.: Optimization for Engineering Design: algorithms and examples. Prentice Hall, New Delhi (1996)
2. Bhandari, V.B.: Design of Machine Elements. McGraw Hill Education (India) Pvt. Ltd., New York (2010)
3. Rao, S.S.: Engineering optimization. New Age International Publishers (1996)
4. Reddy, Y.V.M.: Optimum Design of Belt Drive using Geometric Programming. Journal of Industrial Engineering, 21 (1996)
5. Das, A.K., Pratihar, D.K.: Optimal Design of Machine Elements using a Genetic Algorithms. Journal of Institution of Engineers 83, 97–104 (2002)
6. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
7. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report-TR06. Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
8. Basturk, B., Karaboga, D.: An artificial bee colony (ABC) algorithm for numeric function optimization. In: IEEE Swarm Intelligence Symposium (2006)
9. Dorigo, M., Stutzle, T.: Ant colony optimization. MIT Press (2004)
10. Blum, C.: Ant colony optimization: introduction and recent trends. Physics of Life Reviews 2, 353–373 (2005)
11. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. In: Computer Methods in Applied Mechanics and Engineering, vol. 194, pp. 3902–3933 (2004)
12. Ahrari, A., Atai, A.A.: Grenade explosion method - A novel tool for optimization of multimodal functions. Applied Soft Computing 10(4), 1132–1140 (2010)
13. Kennedy, V., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
14. Clerc, M.: Particle swarm optimization. ISTE Publishing Company (2006)
15. Clerc, M., Kennedy, J.F.: The particle swarm: Explosion, stability and convergence in a multidimensional Complex space. IEEE Transactions on Evolutionary Computation 6(1), 58–73 (2002)
16. Jerald, J., Asokan, P., Prabakaran, G., Saravanan, R.: Scheduling optimisation of flexible manufacturing systems using particle swarm optimisation algorithm. Int. J. Adv. Manuf. Technol. 25, 964–971 (2005)
17. Noorul Haq, A., Sivakumar, K., Saravanan, R., Karthikeyan, K.: Particle swarm optimization (PSO) algorithm for optimal machining allocation of clutch assembly. Int. J. Adv. Manuf. Technol. 27, 865–869 (2006)

18. Saravanan, R., Siva Sankar, R., Asokan, P., Vijayakumar, K., Prabhakaran, G.: Optimization of cutting conditions during continuous finished profile machining using non-traditional techniques. *Int. J. Adv. Manuf. Technol.* 26, 30–40 (2005)
19. Deb, K.: An efficient constraint handling method for genetic algorithm. *Computer Methods in Applied Mechanics and Engineering* 186, 311–338 (2000)
20. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, 341–359 (1997)
21. Gnanambal, K., Babulal, C.K.: Maximum loadability limit of power system using hybrid differential evolution with particle swarm optimization. *Electrical Power and Energy Systems* 43, 150–155 (2012)

Optimal Sizing for Stand-Alone Hybrid PV-WIND Power Supply System Using PSO

D. Suchitra, R. Jegatheesan, M. Umamaheswara Reddy, and T.J. Deepika

Department of Electrical and Electronics Engineering,
SRM University, Kattankulathur, Chennai, Tamilnadu, India
such1978@yahoo.com, uma_reddy_33@yahoo.co.in,
deepikatj@gmail.com

Abstract. Limited fuel reserves and harmful effects of environmental pollution have brought about a lot of development in renewable energy sources. Among these renewable energy sources, solar and wind are found to be the most viable energy sources in the world. Due to their uncertainties in supply and the non-linear characteristics of some of their components, the system becomes unreliable. But when applied as hybrid system, these sources are found to be more economical and efficient. In optimizing the sizing of such hybrid system parameters the evolutionary algorithms have been proved to give faster and more efficient results as compared to classical methods. This work concentrates in finding an optimal configuration of PV modules, Wind turbines and Battery numbers by minimizing the annualized cost considering the loss of power supply probability using Particle Swarm Optimization technique. The radiation and wind data of India obtained from the NASA meteorological website are considered for analysis.

Keywords: Optimum system sizing, loss of power load probability, annualized cost of system.

1 Introduction

Since the oil crisis in the early 1970s, utilization of solar and wind power has become increasingly significant, attractive and cost-effective. In recent years, hybrid PV/wind system (HPWS) has viable alternatives to meet environmental protection requirement and electricity demands. Since they can offer a high reliability of power supply, their applications and investigations gain more concerns now-a-days [1]. Due to the stochastic behavior of both solar and wind energy, the major aspects in the design of the hybrid PV/wind system are the reliable power supply of the consumer under varying atmospheric conditions and the cost of the kWh of energy. To use solar and wind energy resources more efficiently and economically, the optimal sizing of hybrid PV/wind system with battery plays an important role in this respect. In this paper, one optimal sizing model for stand-alone hybrid solar–wind system employing battery banks is developed based on the concepts of loss of power supply probability (LPSP) and annualized cost of system (ACS). The decision variables included in the

optimization process are the PV module number, PV module slope angle, wind turbine number, and also the wind turbine installation height as well as the battery number. The configuration that meets the system reliability requirements with minimum cost can be obtained by an optimization technique – the particle swarm optimization (PSO), which is generally robust in finding global optimal solutions, particularly in multi-modal and multi-objective optimization problems.

2 Model of Hybrid Solar-Wind System

A hybrid solar–wind power generation system consists of PV array, wind turbine, battery bank, inverter, controller, and other accessory devices and cables. A schematic diagram of the basic hybrid system is shown in Fig.1. The PV array and wind turbine work together to satisfy the load demand. When energy sources (solar and wind energy) are abundant, the generated power, after satisfying the load demand, will be supplied to feed the battery until it is charged. On the contrary, when energy sources are poor, the battery will release energy to assist the PV array and wind turbine to cover the load requirements until the storage is depleted. In order to predict the hybrid system performance, individual components need to be modeled first and then their mix can be evaluated to meet the load demand.

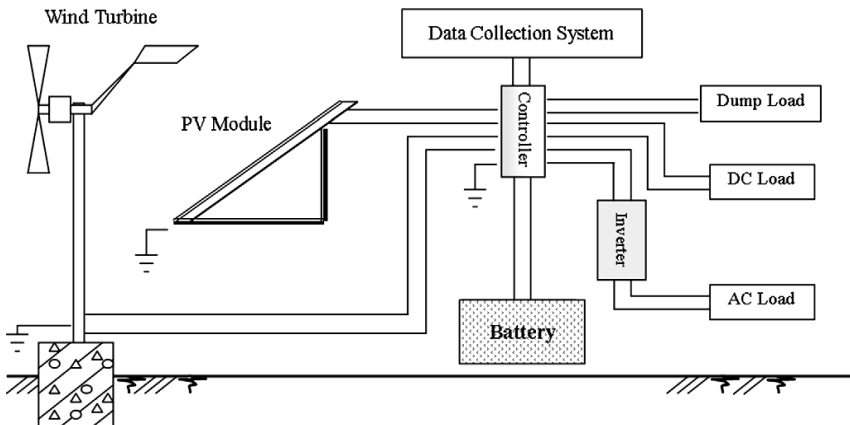


Fig. 1. Block diagram of the hybrid solar–wind system

2.1 PV Array Model

In this paper, a mathematical model for estimating the power output of PV modules is used. The estimation is carried out using a computer program which uses a subroutine for determining the power output of a PV module. Using the solar radiation available on the tilted surface, the ambient temperature and the manufacturers' data for the PV modules as model inputs, the power output of the PV generator, P_{PV} , can be calculated according to equations 1 and 2.

$$P_{\text{module}} = \frac{\frac{V_{oc}}{nKT/q} - \ln\left(\frac{V_{oc}}{nKT/q} + 0.72\right)}{1 + \frac{V_{oc}}{nKT/q}} \cdot \left(1 - \frac{R_s}{V_{oc}/I_{sc}}\right) \cdot I_{sc} \cdot \left(\frac{G}{G_o}\right)^\alpha \cdot \frac{V_{oc}}{1 + \beta \ln\left(\frac{G_o}{G}\right)} \cdot \left(\frac{T_o}{T}\right)^\gamma \quad (1)$$

Where V_{oc} is the open circuit voltage, n is the ideality factor, which varies as $1 < n < 2$, $K = 1.38 \times 10^{-23} \text{J/K}$ is the Boltzmann constant, T is the temperature in Kelvin, T_o is the standard temperature, $q = 1.6 \times 10^{-23} \text{C}$ is the magnitude of the electron charge, R_s is the series resistance, I_{sc} is the short circuit current, I_{sco} is the standard short circuit current, V_{oc} is the standard open circuit voltage, G is the solar radiation W/m^2 , G_o is the standard solar radiation, α is the factor responsible for all the non-linear effects that the photocurrent depends on, β is a PV module technology specific-related dimensionless coefficient and γ is the factor considering all the non-linear temperature–voltage effects[2].

PV modules represent the fundamental power conversion unit of a PV system. It is mandatory to connect PV modules in series and in parallel in order to scale-up the voltage and current to tailor the PV array output. If a matrix of $N_s \times N_p$ PV modules is considered, the maximum power output of the PV system can be calculated by

$$P_{pv} = N_p \cdot N_s \cdot P_{\text{module}} \cdot \eta_{\text{MPPT}} \cdot \eta_{\text{oth}} \quad (2)$$

Where η_{MPPT} is efficiency of the maximum power point tracking, although it is variable according to different working conditions, a constant value of 95% is assumed to simplify the calculations. η_{oth} is the factor representing the other losses such as the loss caused by cable resistance and accumulative dust, etc.

2.2 Solar Radiation on PV Module Surface

The PV module can be placed at any orientation and at any slope angle, but most local observatories only provide solar radiation data on a horizontal plane. Thus, an estimate of the total solar radiation incident on the PV module surface is needed [3]. Generally, the total solar radiation on a tilted surface is calculated by adding the beam, diffuse and reflected solar radiation components on the tilted surface:

$$G_{\text{tt}} = G_{\text{bt}} + G_{\text{dt}} + G_{\text{re}} \quad (3)$$

Where G_{tt} is the total solar radiation on a tilt surface; G_{bt} , G_{dt} and G_{re} are the beam, diffuse and reflected radiation on the tilt surface [4].

Table 1. Data requirements for parameter estimation

| Temperature | G_0 | G_1 |
|-------------|------------------------------------|------------------------------------|
| T_0 | $I_{sc}, V_{oc}, I_{mpp}, V_{mpp}$ | $I_{sc}, V_{oc}, I_{mpp}, V_{mpp}$ |
| T_1 | Null | V_{oc} |

Table 2. Parameter estimation results for the PV module performance

| | α | β | γ | N_{mpp} | $Rs(\Omega)$ |
|------|----------|---------|----------|-----------|--------------|
| Item | 1.21 | 0.058 | 1.15 | 1.17 | 0.012 |

3 Wind Turbine Model

Choosing a suitable model is very important for the wind turbine power simulations. For a typical wind turbine, the power output characteristic can be assumed in such a way that it starts generating at the cut-in wind speed V_C , the power output increases linearly as the wind speed increases from V_C to the rated wind speed V_R , The rated power P_R is produced when the wind speed varies from V_R to the cut-out wind speed V_F at which the wind turbine will be shut down for safety considerations. Then the wind turbine power output can be simulated by

$$P_w (V) = \begin{cases} P_R \cdot \frac{V - V_C}{V_R - V_C} & (V_C \leq V \leq V_R) \\ P_R & (V_R \leq V \leq V_F) \\ 0 & (V \leq V_C, V \geq V_F) \end{cases} \quad (4)$$

The power curve of a wind turbine is nonlinear, the data is available from the manufacturer, and can be easily digitized and the resulting table can be used to simulate the wind turbine performance. Wind speed changes with height and the available wind data at different sites are normally measured at different height levels. The wind power law has been recognized as a useful tool to transfer the anemometer data recorded at certain levels to the desired hub centre

$$V = V_r \left(\frac{H_{WT}}{H_r} \right)^\zeta \quad (5)$$

Where V is the wind speed at the wind turbine height H_{wt} m/s; V_r is the wind speed measured at the reference height H_r m/s; and the parameter ζ is the wind speed power law coefficient. The value of the coefficient varies from less than 0.10 for very flat land, water or ice to more than 0.25 for heavily forested landscapes. The one-seventh power law (0.14) is a good reference number for relatively flat surfaces such as the open terrain of grasslands away from tall trees or buildings [5].

4 Battery Model

Lead-acid batteries used in hybrid solar–wind systems operate under very specific conditions, which is often very difficult to predict when energy will be extracted from or supplied to the battery. Usually, a lead-acid battery is mainly characterized by two indexes, i.e. the state of charge (SOC) and the floating charge voltage.

4.1 Battery State-Of-Charge (SOC)

For a perfect knowledge of the real SOC of a battery, it is necessary to know the initial SOC, the charge or discharge time and the current. However, most storage systems are not ideal, losses occur during charging and discharging and also during storing periods. Taking these factors into account, the SOC of the battery at time $t + 1$ can be simply calculated by [6].

$$SOC(t+1)=SOC(t).\left(1-\frac{\sigma\Delta t}{24}\right)+\frac{I_{bat}(t).\Delta t.\eta_{bat}}{C'_{bat}} \tag{6}$$

Where σ is the self-discharge rate which depends on the accumulated charge and the battery state of health- C' at is the nominal capacity of the battery, η_{bat} is the battery charging and discharging efficiency. It is difficult to measure separate charging and discharging, so manufacturers usually specify roundtrip efficiency. In this paper, the battery charge efficiency is set equal to the round-trip efficiency, and the discharge efficiency is equal to 1. The current rate of the battery at time t for the hybrid solar–wind system can be described by equation 7.

The inverter efficiency $\eta_{inverter}$ is considered to be 92% according to the load profile and the specifications of the inverter. In this case, the wind turbine is assumed to have DC output, so the use of a rectifier is not necessary. But if the wind turbine is designed to connect to an AC grid, then the rectifier losses should be considered for the part of wind energy that has been rectified from AC to DC.

$$I_{bat}(t)=\frac{P_{PV}(t)-\frac{P_{ACload}(t)}{\eta_{inverter}}-P_{DCload}(t)}{V_{bat}(t)} \tag{7}$$

5 Reliability Model Based on LPSP Concept

Several approaches are used to achieve the optimal configurations of hybrid systems in term of technical analysis. In this study, the technical sizing model for the HPWS is developed according to the concept of LPSP to evaluate the reliability of hybrid systems. Loss of power supply probability (LPSP) is defined as the probability that an

insufficient power supply results when the hybrid system (PV array, wind turbine and battery storage) is unable to satisfy the load demand. Two approaches exist for the application of LPSP in designing stand-alone hybrid system. The first one is based on chronological simulation. This approach is computationally burdensome and requires the availability of data spanning a certain period of time. The second approach uses probabilistic techniques to incorporate the fluctuating nature of the resource and the load, thus eliminating the need for time-series data. Considering the energy accumulation effect of the battery, to present the system working conditions more precisely, the chronological method is employed. The power supplied from the hybrid system can be expressed by

$$P_{\text{available}}(t) = P_{\text{PV}} + P_{\text{Wind}} + C \cdot V_{\text{bat}} \cdot \text{Min} \left[I_{\text{bat,max}} = \frac{0.2C'_{\text{bat}}}{\Delta t}, \frac{C'_{\text{bat}} \cdot (\text{SOC}(t) - \text{SOC}_{\text{min}})}{\Delta t} \right] \quad (8)$$

Where C is a constant, 0 indicates battery charging process and 1 indicates battery discharging process. Using the above developed objective function according to the LPSP technique, for a given LPSP value for one year, a set of system configurations, which satisfy the system power reliability requirements, can be obtained. Accordingly, the power needed by the load side can be expressed as:

$$P_{\text{needed}}(t) = \frac{P_{\text{ACload}}(t)}{\eta_{\text{inverter}}(t)} + P_{\text{DCload}}(t) \quad (9)$$

The objective function of the LPSP from time 0 to T can be described by:

$$\text{LPSP} = \frac{\sum_{t=0}^T \text{Power.failure time} (P_{\text{available}}(t) < P_{\text{needed}}(t))}{N_h} \quad (10)$$

6 Economic Model Based on ACS Concept

The economical approach, according to the concept of annualized cost of system (ACS), is developed to be the best benchmark of system cost analysis[1][25]. According to the studied hybrid solar–wind system, the ACS is composed of the annualized capital cost C_{acap} , the annualized replacement cost C_{arep} and the annualized maintenance cost C_{amain} . Then the ACS can be expressed accordingly by (four main parts are considered: PV array, wind turbine, battery and wind turbine tower)

$$\text{ACS} = C_{\text{acap}} (PV + \text{Wind} + \text{Bat} + \text{Others}) + C_{\text{arep}} (\text{Bat}) + C_{\text{amain}} (PV + \text{Wind} + \text{Bat} + \text{Others}) \quad (11)$$

6.1 Annualized Capital Cost

The annualized capital cost of each component (PV array, wind turbine, battery and wind turbine tower) is

$$C_{\text{acap}} = C_{\text{cap}} \cdot \text{CRF}(i, Y_{\text{proj}}) = C_{\text{cap}} \cdot \frac{i \cdot (1+i)^{Y_{\text{proj}}}}{(1+i)^{Y_{\text{proj}}} - 1} \quad (12)$$

Where C_{cap} is the initial capital cost of each component, US\$; Y_{proj} is the component lifetime, year; CRF is the capital recovery factor, a ratio to calculate the present value of an annuity (a series of equal annual cash flows). The annual real interest rate i is related to the nominal interest rate i' and the annual inflation rate f .

6.2 Annualized Replacement Cost

The annualized replacement cost is the annualized value of all the replacement costs occurring throughout the lifetime of the project and it is expressed as

$$C_{\text{arep}} = C_{\text{rep}} \cdot \text{SFF}(i, Y_{\text{rep}}) = C_{\text{rep}} \cdot \frac{i}{(1+i)^{Y_{\text{rep}}} - 1} \quad (13)$$

Where C_{rep} is the replacement cost of the component (battery), US\$; Y_{rep} is the component (battery) lifetime, year; SFF is the sinking fund factor, a ratio to calculate the future value of a series of equal annual cash flows.

6.3 Annualized Maintenance Cost

The system maintenance cost is deemed to be constant every year. The configuration with the lowest ACS is taken as the optimal one from the configurations that can guarantee the required reliability of power supply.

7 Particle Swarm Optimization

7.1 Introduction

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations, the particles are "flown" through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) that it has achieved so far. This implies that each particle has a memory, which allows it to remember the best position on the feasible search space that it has ever visited. This value is commonly called *pbest*. Another best value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the neighborhood of the particle. This location is commonly called *gbest*. The basic concept behind the PSO technique consists of changing the velocity (or accelerating) of each particle toward its *pbest* and the *gbest* positions at each time step. This means that each particle tries to modify its current position and velocity according to the distance between its current position and *pbest*, and the distance between its current position and *gbest*.

7.2 PSO Algorithm

Step 1: Generation of initial condition of each agent. Initial searching points (S_i^0) and velocities (V_i^0) of each agent are usually generated randomly within the allowable range. The current searching point is set to *pbest* for each agent. The best evaluated value of *pbest* is set to *gbest*, and the agent number with the best value is stored.

Step 2: Evaluation of searching point of each agent. The objective function value is calculated for each agent. If the value is better than the current *pbest* of the agent, the *pbest* value is replaced by the current value. If the best value of *pbest* is better than the current *gbest*, *gbest* is replaced by the best value and the agent number with the best value is stored.

Step 3: Modification of each searching point. The current searching point of each agent is changed using follow equation.

$$V_i^{k+1} = V_i^k + C_1 \times \text{rand}()_1 \times (pbest_i - S_i^k) + C_2 \times \text{rand}()_2 \times (gbest - S_i^k) \quad (14)$$

$$S_i^{k+1} = S_i^k + V_i^{k+1} \quad (15)$$

Where

V_i^{k+1} is the Velocity of particle i at iteration $k+1$; V_i^k is the Velocity of particle i at iteration k ; S_i^{k+1} is the Position of particle i at iteration $k+1$; S_i^k is the Velocity of particle i at iteration k ; C_1 is the Constant weighing factor related to *pbest*; C_2 is the Constant weighing factor related to *gbest*; $\text{rand}()_1$ $\text{rand}()_2$ are Random numbers between 0 and 1; $pbest_i$ is the best Position of particle i and *gbest* is the best Position of the swarm.

Step 4: Checking the exit condition. The current iteration number reaches the predetermined maximum iteration number, then exits. Otherwise, the process proceeds to Step 2.

8 System Optimization Model with PSO

The minimization of ACS function is implemented employing PSO, which can dynamically searches for the optimal configuration. The decision variables included in the optimization process are the PV module number N_{PV} , wind turbine number N_{WT} , battery number N_{bat} , PV module slope angle β' and wind turbine installation height H_{WT} . The initial assumption of system configuration will be subjected to the following inequalities constraints:

$$\text{Min}(N_{PV}, N_{WT}, N_{bat}) \geq 0 \quad (16)$$

$$H_{low} \leq H_{wt} \leq H_{high} \quad (17)$$

$$0 \leq \beta' \leq 90 \quad (18)$$

The hourly data used in the model are the solar radiation on horizontal surface, ambient air temperature, the wind speed and the load power consumption on an annual basis. The year 2001 is chosen as the Example Weather Year in Chennai, India to represent the climatic conditions for the studied project design in the following optimization process. The PV array power output is calculated according to the PV system model by using the specifications of the PV module as well as the solar radiation data. The wind turbine performance calculations need to take into account the effects of wind turbine installation height. The battery bank, with total nominal capacity C'_{bat} (Ah), is permitted to discharge up to a limit defined by the maximum depth of discharge DOD, which is specified by the system designer at the beginning of the optimal sizing process. The system configuration will then be optimized by employing particle swarm optimization, which dynamically generates the optimal configuration to minimize the annualized cost of system (ACS). For each system configuration, the system's LPSP will be examined for whether the load requirement (LPSP target) can be satisfied. So, for the desired LPSP value, the optimal configuration can be identified both technically and economically.

9 Result and Discussion

9.1 The Hybrid Solar-Wind Project Description

The proposed method has been applied to design a hybrid system to meet a 1500W (24 V AC) and 300W (24 V DC) load in a location at Chennai, Tamilnadu in India.

According to the project requirement and technical considerations, continuous 1800W energy consumption is assumed as the constant demand throughout the period. The technical characteristics of the PV module and the wind turbine used in the studied project are listed in Tables 3 and 4. The GFM-1000 lead-acid batteries are employed in the project. They are specially designed for deep cyclic operation in consumer applications like the hybrid solar–wind energy systems. Each set of batteries has twelve 2 V/1000 Ah cells which are connected in series to give a nominal output voltage of 24 V. The initial capital cost, replacement cost, maintenance cost and lifetime of each component are in Table 5.

Table 3. Specifications of PV module

| V_{OC} (V) | I_{SC} (A) | V_{max} (V) | I_{max} (A) | P_{module} (W) |
|--------------|--------------|---------------|---------------|------------------|
| 21 | 6.5 | 17 | 5.73 | 100 |

Table 4. Specifications of wind turbine

| Rated Power (kW) | Cut in speed V_c (m/s) | Rated speed V_r (m/s) | Cut-off speed V_f (m/s) | Tower Hhigh (m) | Tower Hlow (m) |
|------------------|--------------------------|-------------------------|---------------------------|-----------------|----------------|
| 6 | 2.5 | 10 | 25 | 50 | 10 |

Table 5. The costs and lifetime aspect for the system components

| Components | Initial Cost (US \$) | CapitalReplacement Cost (US \$/kAh) | Maintenance Cost in the 1 st year (US \$) | Life time (years) | Interest rate I' (%) | Inflation rate f (%) |
|--------------|----------------------|-------------------------------------|--|-------------------|----------------------|----------------------|
| PV array | 6500 | Null | 65 | 25 | 3.75 | 1.5 |
| Wind turbine | 3500 | Null | 95 | 25 | | |
| Battery | 1500 | 1500 | 50 | Null | | |
| Tower | 250 | Null | 6.5 | 25 | | |
| Others | 8000 | Null | 80 | 25 | | |

Table 6. Optimal sizing results for the hybrid solar-wind system

| N_{PV} | N_{wt} | N_{bat} | H_{wt} (m) | Cost (\$) | LPSP | β' |
|----------|----------|-----------|--------------|-----------|------|----------|
| 130 | 3 | 6 | 34.89 | 14750 | 0 | 24° |

9.2 Power Reliability and Annualized Cost of System

The hourly radiation data for PV system and hourly wind speed data for wind turbine modeling for a location at Chennai in India is shown in fig 2 and 3. The temperature data required for the simulation of the PV module is assumed based on the meteorological data corresponding to this radiation input. The output obtained for the

considered hybrid system executed in MATLAB environment has been shown in Table 6. The minimum annualized costs of system for different LPSP (power reliability requirements) are calculated by the proposed optimal sizing method. It is observed that higher power reliable systems are more expensive than lower requirement systems. Choosing an optimal system configuration according to system power reliability requirements can help save investments and avoid blind capital spending.

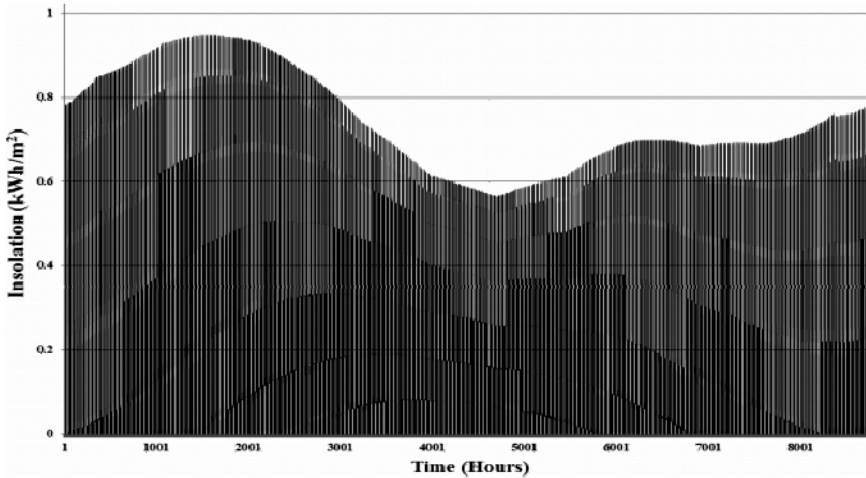


Fig. 2. Hourly radiation input data for Chennai

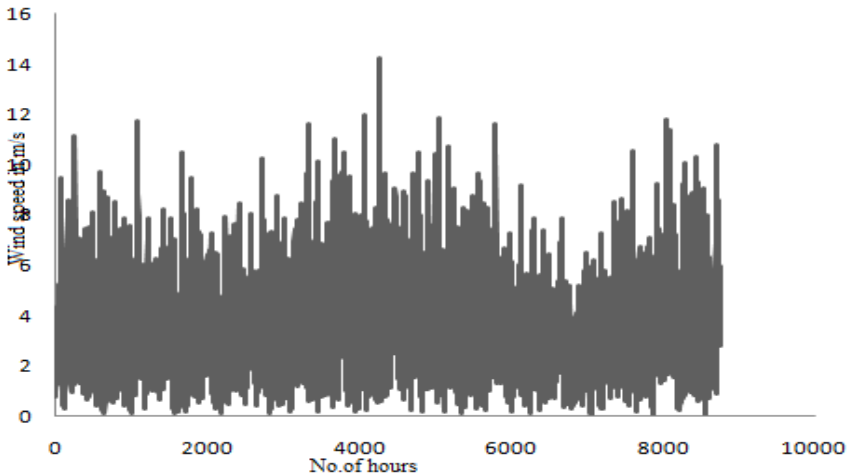


Fig. 3. Hourly wind input data for Chennai

10 Conclusion

Power supply reliability under varying weather conditions and the corresponding system cost are the two major concerns in designing PV and wind turbine systems. In order to utilize renewable energy resources of both solar and wind energy efficiently and economically, an optimal sizing method is developed in this work based on particle swarm optimization (PSO). The PSO based optimization model developed in this paper, optimizes the sizing of autonomous hybrid PV-wind power generation system using a battery bank as a backup source. The model has been used to calculate the system optimum configuration which can achieve the desired loss of power supply probability (LPSP) with minimum annualized cost of system. The decision variables included in the optimization process are the PV module number, wind turbine number, battery number, PV module slope angle and wind turbine installation height. Good optimal sizing performance of the algorithms has been found, and the optimal solution is a hybrid solar–wind system with a battery as a backup source. The accuracy of the optimal solution can further be improved using advanced or Hybrid PSO algorithm. The loss of power supply probability is highly influenced by the weather conditions of that year, so under the bad weather conditions, the probability of loss of power supply will be higher. To meet out these conditions, the renewable energy sources have to be oversized.

References

1. Yang, H.X., Lu, L., Burnett, J.: Weather Data and Probability Analysis of Hybrid Photovoltaic-Wind Power Generation Systems in Hong Kong. *Renewable Energy* 28, 1813–1824 (2003)
2. Van Dyk, E.E., et al.: Long-Term Monitoring of Photovoltaic Devices. *Renewable Energy* 22, 183–197 (2002)
3. Duffie, J.A., Beckman, W.A.: *Solar Engineering of Thermal Process*. John Wiley & Sons, USA (1980)
4. Luis, C., Sivestre, S.: *Modeling Photovoltaic Systems Using PSpice*. John Wiley & Sons Ltd., Chichester (2002)
5. Gipe, P.: *Wind Energy Comes of Age*, p. 536. John Wiley & Sons (1995)
6. Berndt, D.: *Maintenance-Free Batteries*. John Wiley & Sons, England (1994)
7. Ashari, M., Nayar, C.V.: An Optimum Dispatch Strategy Using Set Points for A Photovoltaic (PV)–Diesel–Battery Hybrid Power System. *Solar Energy* 66(1), 1–9 (1999)
8. Kellogg, W., Nehrir, M.H., Venkataramanan, G., Gerez, V.: Optimal Unit Sizing for A Hybrid PV/Wind Generating System. *Electric Power System Research* 39, 35–38 (1996)
9. Gavanidou, E.S., Bakirtzis, A.G.: Design of A Stand Alone System With Renewable Energy Sources Using Trade Off Methods. *IEEE Transactions on Energy Conversion* 7(1), 42–48 (1993)
10. Yang Lu, L., Burnett, J.H.X.: Investigation on Wind Power Potential on Hong Kong Islands–An Analysis of Wind Power and Wind Turbine Characteristics. *Renewable Energy* 27, 1–12 (2002)

11. Borowy, B. S., Salameh, Z.M.: Methodology for Optimally Sizing The Combination of A Battery Bank and PV Array in A Wind/PV Hybrid System. *IEEE Transactions Energy Conversion* 11(2), 367–373 (1996)
12. Yang, H.X., Lu, L., Zhou, W.: A Novel Optimization Sizing Model for Hybrid Solar-Wind Power Generation System. *Solar Energy* 81(1), 76–84 (2007)
13. Zhou, W., Yang, H.X., Fang, Z.H.: A Novel Model for Photovoltaic Array Performance Prediction. *Applied Energy* 84, 1187–1198 (2007)
14. Kattakayam, T.A., Srinivasan, K.: Lead Acid Batteries in Solar Refrigeration Systems. *Renewable Energy* 29(8), 1243–1250 (2004)
15. Eftichios, K., et al.: Methodology for Optimal Sizing of Stand-Alone Photovoltaic/Wind-Generator Systems Using Genetic Algorithms. *Solar Energy* 80, 1072–1188 (2006)
16. Guasch, D., Silvestre, S.: Dynamic Battery Model for Photovoltaic Applications. *Progress in Photovoltaics: Research and Applications* 11, 193–206 (2003)
17. Markvart, T.: Sizing of Hybrid PV-Wind Energy Systems. *Solar Energy* 59(4), 277–281 (1996)
18. Schott, T.: Operational Temperatures of PV Modules. In: 6th PV Solar Energy Conference, pp. 392–396 (1985)
19. Tina, G., Gagliano, S., Raiti, S.: Hybrid Solar/Wind Power System Probabilistic Modelling for Long-Term Performance Assessment. *Solar Energy* 80, 578–588
20. Yang, H.X., Lu, L.: Study on Typical Meteorological Years and Their Effect on Building Energy and Re-newable Energy Simulations. *ASHRAE Transactions* 110(2), 424–431 (2004)
21. Abouzahr, I., Ramakumar, R.: Loss of Power Supply Probability of Stand-Alone Electric Conversion Systems: A Closed Form Solution Approach. *IEEE Transactions on Energy Conversion* 5, 445–452 (1990)
22. Borowy, B.S., Salameh, Z.M.: Optimum Photovoltaic Array Size for A Hybrid Wind/PV System. *IEEE Transactions on Energy Conversion* 9, 482–488 (1994)
23. Lu, L., Yang, H.X., Burnett, J.: Investigation on Wind Power Potential on Hong Kong Islands – An Analysis of Wind Power and Wind Turbine Characteristics. *Renewable Energy* 27, 1–12 (2002)
24. Yoon-Ho, K., Hoi-Doo, H.: Design of Interface Circuits with Electrical Battery Models. *IEEE Transactions on Industrial Electronics* (1997)
25. Ould Bilal, B., Sammbou, 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 the minimization of the loss of power supply probability(LPSP). *Renewable Energy* 35, 2388–2390 (2010)

A Peer-to-Peer Dynamic Single Objective Particle Swarm Optimizer

Hrishikesh Dewan^{1,2}, Raksha B Nayak² and V Susheela Devi¹

¹ Department of Computer Science & Automation
Indian Institute of Science, Bangalore
{hrishikesh.dewan,susheela}@csa.iisc.ernet.in

² Knowledge & Innovation
Siemens Corporate Technology & Development Center, Bangalore,
raksha.nayak@siemens.com

Abstract. Natural problems are not static but dynamic in nature. Dynamism could be due to a large number of factors such as inability to gauge accurate objective functions or changing constraints in a random environment. As such, dynamic optimization problems are an important requirement for solving a number of real life optimization problems. In this paper, we present a novel technique that not only detects the changing environment but also repositions itself intelligently in the changed landscape. Our algorithm is parallel and works on top of distributed peer-to-peer network in a complete asynchrony. We simulate our results, in a distributed environment ranging from 10 to 100 machines and show the scalability to be able to increase to thousands of nodes if required. The algorithm is tested on a popular set of benchmark functions and we show/compare the results with known solutions of such types.

1 Introduction

Dynamic single-objective problems (DSOP) are an important class of optimization problems wherein the decision variables or the objective function change over a period of time. The challenge in DSOP is to track the changing global minimum/maximum. Formally, a DSOP can be defined as in [1]. There are several researches that have been conducted in solving DSOP [1–3]. All these algorithms work in a single computing system where the algorithm spans threads by taking advantage of the multiple cores as present in the system. However, with the increase in the number of variables, dimension of variables and/or objective function, such algorithms will take a large amount of time to find the global minima. One extension of these parallel threads is to use a dedicated cluster of machines. A dedicated cluster of machines with almost homogenous hardware/software and internetwork is however costly and requires dedicated maintenance.

$$\begin{aligned} \text{Minimize/Maximize} \quad & f_i(x, t), \quad \text{where} \quad x = (x_1, x_2, x_3 \cdots x_n) \\ & g_j(x, t) \leq 0, \quad j = 1, 2, 3 \cdots J \\ & h_{k,t}(x) = 0, \quad k = 1, 2, 3, \cdots K \\ & x_i = [\text{low}_i, \text{high}_i] \end{aligned} \tag{1}$$

On the other hand, peer-to-peer distributed systems (P2P) provide a viable alternative where individual machines can participate in a computationally rich optimization problem without any adherence to the strict administrative control as that of a cluster. A P2P system is characterized by a loose connection of independent machines using an overlay network where nodes can join and leave at their own wish. Being a loose cluster of machines, the design and use of a P2P system for optimization however has a number of challenges. First, in a P2P network there is no guarantee of node availability. A node may join, contribute for a brief period and exit altogether from the network. Such ungraceful leave of a node could be due to intentional leaving, software error or network partition in the P2P network. Further, even if a node is present, due to heterogeneity or variance in the configuration of hardware and software capability among the peer nodes, there could be an unequal distribution of job across the systems. Both of the above problems are crucial from the purview of a computationally rich job such as an optimization algorithm. The solution to the first problem of non-availability of nodes is important because without it an optimization problem will never be complete and the solution to the heterogeneity is important because without it, there would be unequal distribution of load. Unequal distribution may lead to inefficient utilization of the resources as present in the P2P system. Thus a P2P system modeled for solving optimization problems does not only have to ensure availability of nodes for continuous optimization of the problem but should also have the provision of completely utilizing the resources of the system.

In this paper, we present such a P2P system that aims to accurately track and find global minimum/maximum for a DSOP. Our algorithm is based on Particle Swarm Optimization (PSO) algorithm. We have modified certain aspects of the PSO algorithm for efficient detection of change in the environment and move towards the changing global minimum/maximum. Our algorithm consists of a number of sub-swarms, where each node runs a single swarm optimization method. Sub-swarms connect to other sub-swarms with a deterministic and a random method thereby facilitating both exploitation and exploration. We also handle fault tolerance of nodes comprising the system and distribute load evenly across the system. The algorithm has been tested with a wide range of available benchmark functions and in all cases we have achieved results which are at par with the single computing systems. The architecture of P2P network is elastic and it can scale well to thousands of nodes without any significant reduction of performance. Although there is a large body of work in the areas of parallel particle swarm optimization and such other swarm intelligence techniques, there is, however, no such work that directly tries to optimize a DSOP in a P2P network.

This paper is organized as follows. In section 2, we define the related work that compares our work with the existing body of research. In section 3, we define the P2P network and the algorithms that are used to solve the DSOP. In section 4, we show the results of the experiments and analysis and finally in section 5, we conclude the paper and provide the directions of future research in this area.

2 Related Work

There is not much work done on the DSOP and most of these works are based on algorithms that run in a single system. For example, the algorithms as defined in [3–5] are examples of single system based optimization algorithms. [6] is a survey of well known algorithms in the areas of evolutionary computing that have been developed so far to solve dynamic optimization problems. [7, 8] are also some of the recent works in this area where dynamic optimization problems were solved. However, these developments are single system based and they have not been tried or tested in distributed P2P systems. Our paper relates to finding the global minimum/maximum using a set of loosely connected P2P systems. Hence, a direct comparison of our results with the results of existing techniques is not feasible.

3 Architecture and Algorithm

The entire distributed P2P framework for DSOP can be divided into two stages. The first stage is the network architecture and algorithms that are used for creating and maintaining the network. The second stage is the algorithm used for tracking/computing the global minimum/maximum of the optimization problem. The first part is the middleware that is involved in keeping the distributed systems intact while the second part is the application itself that runs on top of this distributed system. Since swarm intelligence requires information dissipation, the routing table of the underlying overlay network of the distributed system is partially populated and guided by the application at the top. We first define the network and then the actual optimization algorithm in the subsequent sections.

3.1 Network Architecture and Algorithms

We use an overlay network of nodes for communication of system as well as application level messages. The overlay network is a circular ring and is formed by applying consistent hashing functions. When a node joins the network, it is provided with a unique 160 bit identifier. The identifier is derived by concatenating time and MAC address of the node as an input to SHA-1 hash function. Based on the output, the ring is organized lexicographically. Figure 1 is an illustration of the same. For routing messages from one node to the other, each node maintains a routing table. The routing table entries are of three different types. They are (i) Lexicographic neighbors (bi-directional) in both the clockwise and anti clock wise directions (a configurable number each in either direction), (ii) unidirectional random links (based on the number of 1s in the identifier) and (iii) objective space neighbors. Objective space neighbors are the areas in decision variable space which are adjacent to the node that is currently being computed. For 2D space as illustrated, the number of neighbors is 4 and in general for

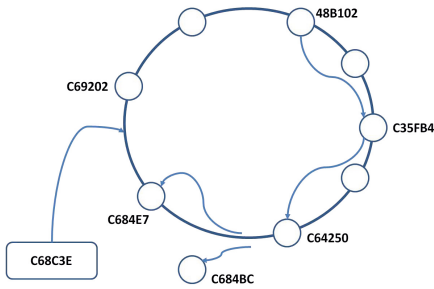


Fig. 1. An overlay network of nodes

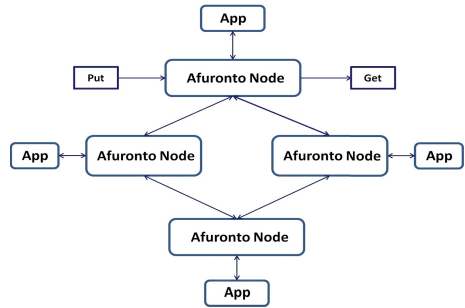


Fig. 2. An overlay network of nodes

n-dimensional space there are $2n$ neighbors. The reason for maintaining the objective space neighbors is to refer to it during the strategic re-work negation algorithm defined in the subsequent sections.

When a node needs to send a message to yet another node, the routing table for the node is consulted to find the nearest node. Nearest node distance is the lexicographic distance. The algorithm then forwards the node to the nearest node and the same procedure is repeated till either the destination node is reached or there is no node of that identifier. In case of the former, the message is consumed by the destination node whereas in case of the latter, the message is either consumed by the most preceding node or an error is returned to the user. Due to non-uniformity of node links, the reverse path can be different from the source node. Also, due to non-uniformity of processing capability, each node may compute widely disparate function spaces. Therefore the neighbor space routing table changes from time to time and is not bound to be static during the entire life cycle of a node. Our network differs from other DHT-based network such as Chord [1], Pastry [2] etc. by including extra neighbor node information related to our objective function. This modification is sought to decrease the message overload and also for fast convergence. When a node completes its job of finding the Pareto front, it does not seek to enter into some other nodes dimensional space. Instead, the completed or the lightly loaded node simply skips the region of space and moves to the space which is slowly progressing or not yet processed. Thus, unlike the traditional PSO algorithms where there is a large amount of duplication of work, this design ensures very little duplication of work. Hence, compared to other algorithms, the relative time to convergence is faster. Due to large number of dimensions, the effective neighbors are roughly around $2n$. This is a large number even for moderately size dimensions. Every node maintains a single indirection to its parent and each parent includes the list of siblings. The number of siblings is not more than the number of dimensions that is being used to divide the hyper-plane. Thus, the maximum number of neighbors that a node has to store is dependent upon the problem statement and also on the factors of division. For example, for three dimensional vectors, we first divide the function space in one of its axis. The second level includes the divisions of the second and so on. This increases the time to propagation but it prohibits the overcrowding of the routing table entries of the node.

3.2 Node Management

In a peer-peer network all nodes are symmetric and equal. Whenever a node has to participate in the function optimization computation, the first operation is the join operation. Upon completion of join, a node acquires a membership to the network and participates in all its activities, which includes function optimization, message passing, load balancing and fault-tolerance. Since each peer-node is responsible for at-least a certain chunk of work, there must be well-defined node leaving protocol. In this section, we define the node joining and leave protocols. It is important to note that in an P2P network with no central administrative unit, nodes can join and leave at any moment. Such abrupt change in the network topology could be due to malfunctioning of hardware, software or the network. Hence, the join and the leave protocol must handle these extreme but major use cases as well. As noted in section 3.1, whenever a node joins, it first creates a new pseudo-random ID of 160 bits. After the ID is created, it then searches for an existing node in the network to communicate with. The list of existing nodes must be supplied to the new node before it starts its operation and is, by and large, a manual process. Once the node is connected to any of the existing nodes, the first message it composes is to find duplication of its own calculated node ID. Such a message is routed to the network using the newly connected existing node. If there are no such nodes, then the node can safely join the network. On the other hand, if such a node exists, the new node creates a new ID and recursively follows the same process. Joining the network involves creating the routing table entries. After the routing table entries are successfully created, the node connects to any node in the network for work units. Since a node already existent in the network is completely participating in the optimization process, the new node requests work from the already existent nodes. In our present protocol, we allow a new node to acquire work from a number of existing nodes and selects the work that seems appropriate. Upon selection of the work unit, the neighbors nodes are identified and the objective neighbor node work is filled. At this stage the routing table entries are complete and also the necessary code and work unit is with the new node. The new node starts executing its own sequential PSO from this point. Note that this type of work selection is completely deterministic and may take some time if the already existing nodes do not have work to allocate to a new node. The other joining protocol that we have investigated is to randomly select a work. Once the new node acquires the necessary code for the functions, its solution space and work division technique, it randomly selects a work unit from the entire available work unit. Since the routing table entries maintains neighbor node information based on the decision space neighbors, a message is sent from the new node to the node which is responsible for the work chunk. If the work chunk is available, then it is allocated to the new node or the node recursively continues the random work unit selection process again until it gets the desired work load. It is important to note that if the number of nodes are larger than the available work units, then neighbor nodes can further partition its work space and allocate a few to the newly joined node.

The node leaving protocol is opposite of what is defined in the node joining protocol. The leaving node broadcast leave messages to all its neighbors and waits for a few seconds before it leaves the network. Upon leaving the network, a hole is created in the overlay network and routing table entries of the neighbor nodes are modified. Dynamic node churns are described in more detail in a later subsection.

3.3 Load Balancing and Fault Tolerance

A P2P network is always a mix of diversified components: diversified in terms of hardware resources and software components available for computation. Therefore, there is no uniformity in the completion time of a solution space. Some nodes may take a long time to compute a work unit whereas some nodes may complete the same in 1/10th of the time taken by the other node. As a result, load balancing of nodes is an important requirement in such a diversified P2P network. We balance loads not instantaneously, but after repeated step intervals. A step interval is a finite number of iterations. After completion of each step, the node propagates its load to the neighbors using a broadcast. Nodes that are lagging behind comparatively are further propagated. For broadcasting, there are two specific rules. For each node we maintain a least and utmost load, which are respectively 20% and 80 % of the load. If the CPU utilization falls below or above this limit, the node broadcasts this information to the neighbors. Every node therefore maintains the load of its neighbors. If, however, the load is not below or above this threshold limit, there is no message sent. Hence, load information table is not as populated as the routing table entries. Once a node receives such information, it compares the load with all the entries and tries to achieve equilibrium by matching low capacity nodes with the high capacity ones. If, on the other hand, there are nodes that are still not yet matched, then the information is passed on to the nodes neighbors. The process is repeated until either there is a match or there are no nodes to match. When a node completes its allocated work unit and there are no more pending works, the computational utilization decreases by 10%. Under this circumstance, the broadcast is sent from the node to all its neighbors till it receives new chunk of data. Thus the network tries to balance load at the neighborhood first and if unsuccessful, propagates the information to the next level. With this, there are no central co-coordinators required and also the number of messages required for balancing the load is less.

As in the case of load balancing, fault tolerance is also handled co-operatively. Each node upon joining the network maintains three fault tolerant connections to three other nodes. These nodes need not be entries in the routing table and random selections from the nod identifier space. After every successful time interval, which is configurable, the node sends the best positions, work unit in allocation information to each of these nodes. Failure to receive updates by the majority, either due to its software/hardware or network partition, signals the node as dead and a new node is selected for execution of the work.

3.4 P2P DSOP

Our optimization algorithm is designed to handle all possible cases of change detection in the environment. When there is no access to the optimization functions, change may happen in multiple different points in the valid search landscape. For example, change could be only minimal and is concentrated near the global optimum or there is a complete change in the optimization landscape where the new global optimum is shifted to a far point from the preceding global optimum.. Further, changes might happen in two different ways - either there is a change in the decision variable space or there is a change in the optimization in the objective space. If the type of change is known apriori, it is relatively trivial to reposition the particles in the changed landscape and find solutions close to the previously known global optimum. Since the change in the environment cannot be explicitly determined, a range of pattern recognition techniques together with time series analysis can be used to predict the change. However, such prediction will require historical data and we refrain from discussing it in this paper. In the future, we would integrate such prediction techniques for better positioning of particles in the changing search landscape. Interested reader may refer to [9, 10] where technique such as these have been applied.

In our present algorithm, we have two different types of particles - new particle types and old particle types. New particle types are like foragers and they wander in the solution space within a fixed radius. Old particles are just like the old particle swarm particles and they follow the leader which has the best values. Every particle stores the location where it achieved the non-dominated point in its memory and after a pre-defined step, it recomputes the non-dominated set once again. This is being done to check if there is a change in the environment. If there is a change in the environment, the change notification is sent to each of its neighbors by a broadcast. The change notifications are high priority messages and they are broadcast recursively. Since, change in the environment can happen at multiple different search spaces and many particles detect it, the number of broadcast messages could be enormously high. To reduce it, we use a threshold. Beyond a certain threshold within a fixed interval, if a particle receives change notifications, further notifications are simply suppressed. Therefore, such change notification only generates a small number of messages. Further, if a particle received a change notification, and thereafter recognizes a change in its landscape within the pre-determined interval, the change notification is also suppressed. The newborn particles do not have any memory and they just report the optimum values.

The rationale behind the introduction of new particles is to detect changes even when the particles have converged to a global optimum. Cases of this type generally occur when the change in the environment is slow. If the particles converge to the global optimum and there is no particular change in their vicinity, the PSO will certainly miss any new updates. This is generally the case with sentry based particles as defined in [2]. The new particles therefore wander around the search space with a goal to find positions which are better than the converged optimum. This also means that the optimization process is non-stop and once started the optimization will continue till it is stopped by intervention. In the distributed system, each

process executes a sub-swarm and the number of particles both old and new can vary. Such variations are necessary as machines themselves have varying configurations. Further as noted in section 3.1 A, each sub-swarm is only responsible for its search landscape. The sub-swarms exchange messages among themselves which includes the gbest of its search space.. The algorithm used inside each sub-swarm is shown in Algorithm 1 and Algorithm 2.

```

Algorithm : Particle Execute
Description : This method is executed by each particle in the PSO
Input : Gbest
Result : Null or particle's own best position
Begin
  changeDetected by this particle  $\leftarrow$  false;
  if (iteration count % curve check factor) $\neq$ 0 then
    | if (best result of this particle  $\neq$  ComputeFunction(pbest of this particle)) then
    | | best result of this particle  $\leftarrow$  null;
    | | pbest of this particle  $\leftarrow$  null;
    | | changedetected by this particle  $\leftarrow$  true;
    | end
  end
  if changeDetected by this particle  $\leftarrow$  false then
  | update velocity;
  | update current position;
  end
  result  $\leftarrow$  ComputeFunction(n-dimensional currentPosition));
  if (result dominates bestresult) then
  | pbest of this particle  $\leftarrow$  current position;
  | bestResult of this particle  $\leftarrow$  result;
  | pBest  $\leftarrow$  pbest of this particle;
  | result  $\leftarrow$  bestResult of this particle;
  | changeDetected  $\leftarrow$  change detected by this particle;
  end
  else return null;
  return (pBest,result,changeDetected);
End

```

Algorithm 1. Execution of a Single Particle

4 Experiment and Results

Generally, there are two most popular benchmark functions that are used for testing DSOP. The first is the moving peaks benchmark as defined in [11] and the second is [12]. In this paper, we have shown the results for only [11]. In [11], there are three different test functions, hill, cone, and slope. We have used cone function. The cone function by default has peaks of 5 dimensions. However in our experiment we have increased the dimensions to 10, 15 and 20. Together with distributed systems framework, the gbest of the current iteration is transferred to the neighbors of a node. Each neighbor runs a sub-swarm. The results of the execution for the cone function are shown in table 2-5. As shown, we have scaled up the dimensions of the peak to 5, 10, 15 and 20. The number of peaks ranges from 5 to 200. Each such setting (particle, number of peaks, dimensions, FES) is run for 25 trial runs. The values as mentioned in the table are the offline error as shown in [11]. Further, in each of the trials, the number of new particle type is 5% of number of particles and the number of particles that are repositioned after the change detection is between 10-50 % randomly selected from the group. The maximum number of children that can be generated by the best particle is 5.

```

Algorithm : Particle Swarm Optimization
Input : Input dataset
Result: Optimized position
Begin
Initialize N particles including old type and new type particles;
Initialize Gbest to null;
while Iteration count < Total number of iterations do
  for p ← 0 to total number of particles do
    (pBest,result,changeDetected) ← ExecuteParticle;
    if result != null then
      if changeDetected==true then
        gbest ← null;
        global non - dominated element ← null;
        Notify other particles about the change;
        Re randomize T% of the old type particles;
      end
    else
      mother particle ← global non - dominated element;
      if (mother particle != new born particle type) then
        Call Give Birth to new born particles() of mother particle;
        Add the new born particle list to the main list;
        Delete the old new born particle list from the main list;
        if (mother particle == new particle type) then
          Re-randomize T% of the old type particles;
        end
      end
    end
  end
  Increment the Iteration count;
end
Non - dominated set is the result;
End

```

Algorithm 2. Sub Swarm Algorithm

Figure [3-6] shows the convergence plot for the functions for a few iterations. As evident from the figure, the theoretical global maxima is closely tracked and followed. Apart from this, we have also executed an experiment for 500 particles distributed over a set of 50 distributed nodes. Each node executes 10 particles and some of the nodes in the setup execute only the newborn particles. Table 2-5 shows the result for the same. It is important to note that in all the other contemporary works in the single system based dynamic single objective optimization, results for large dimensions such as 10,15 and 20 have not been reported. Further, to ascertain the best values for the re-randomization, we have shown the sensitivity analysis of the moving peaks benchmark with 10-50 % re-randomization of the old particle types. Table 1 shows the result. As evident from the table, with the increase in the number of re-randomization particles, the offline error performance degrades. The reason for degradation is due to the fact that the change in the optimum position is very meagre and also near the previously found optimum position. If the change would have drastic and at a different position which is a distance apart, maximum re-randomization would have been helpful. Further, to show the performance of the distributed system, we run an experiment simulating 100 nodes for a 5 peaks , 200 dimension moving peaks problem. The results are shown in Table 6. Compared to the same results for a eight node system with 200 particles in total, the result is better in the same amount of time. The P2P systems are very fruitful when they are used for solving large scale optimization problem. For example, as mentioned in [13], the large scale optimization problems generally takes inordinate of time.

As evident from the results, even with large dimensions, the average offline error is low. If the number of particles used is only 100 and the functions changes

Table 3. Statistical results for 10 dimensions

| FES | (Peaks) | 1 | 5 | 10 | 20 | 30 | 40 | 50 | 100 | 200 |
|-------|------------|------------|------------|------------|------------|------------|------------|------------|------------|-----|
| 500 | 6.13± 2.39 | 6.61± 2.70 | 7.37± 2.71 | 7.39± 3.10 | 7.60 ±3.46 | 7.80 ±3.11 | 7.41 ±2.95 | 8.06± 3.82 | 7.33 ±3.64 | |
| 1000 | 2.04 ±0.41 | 2.21± 0.33 | 2.28 ±0.30 | 2.45± 0.39 | 2.25 ±0.27 | 2.35 ±0.28 | 2.33 ±0.29 | 2.28± 0.33 | 2.03 ±0.24 | |
| 2500 | 0.67 ±0.08 | 1.62 ±0.17 | 1.89 ±0.14 | 1.94± 0.18 | 1.95 ±0.18 | 1.97 ±0.19 | 1.96 ±0.20 | 1.98 ±0.23 | 1.60 ±0.24 | |
| 5000 | 0.28 ±0.4 | 2.05±0.07 | 2.39±0.33 | 2.61±0.23 | 2.54±0.2 | 2.59±0.25 | 2.62±0.2 | 2.65±0.32 | 2.12±0.26 | |
| 10000 | 0.06± 0.01 | 2.04 ±0.26 | 2.53 ±0.28 | 2.65± 0.38 | 2.66± 0.29 | 2.95 ±0.29 | 2.95± 0.34 | 3.03 ±0.32 | 2.32 ±0.68 | |

Table 4. Statistical results for 15 dimensions

| FES | (Peaks) | 1 | 5 | 10 | 20 | 30 | 40 | 50 | 100 | 200 |
|-------|------------|------------|------------|------------|------------|-------------|------------|------------|------------|-----|
| 500 | 22.88±4.37 | 26.55±6.77 | 26.16±7.13 | 28.85±7.61 | 28.17±6.14 | 27.36± 6.19 | 25.75±7.59 | 27.81±6.32 | 27.52±5.84 | |
| 1000 | 3.99±0.72 | 4.33±0.62 | 4.39±0.61 | 4.30±0.75 | 4.19±0.56 | 4.41±1.01 | 4.05±0.44 | 4.06±0.61 | 3.77±0.72 | |
| 2500 | 1.97±0.28 | 2.99±0.39 | 3.13±0.30 | 3.24 ±0.33 | 3.28±0.42 | 3.21±0.28 | 3.33±0.39 | 3.32±0.26 | 3.07±0.28 | |
| 5000 | 1.2±0.09 | 3±0.28 | 3.43±0.33 | 3.61±0.27 | 3.66±0.32 | 3.75±0.31 | 3.73±0.32 | 3.77±0.37 | 2.95±0.42 | |
| 10000 | 0.54± 0.11 | 2.48 ±0.25 | 2.94 ±0.39 | 3.16 ±0.30 | 3.41± 0.47 | 3.43 ±0.38 | 3.42 ±0.36 | 3.38 ±0.35 | 2.54 ±0.50 | |

Table 5. Statistical results for 20 dimensions

| FES | (Peaks) | 1 | 5 | 10 | 20 | 30 | 40 | 50 | 100 | 200 |
|-------|--------------|--------------|--------------|--------------|-------------|-------------|------------|-------------|-------------|-----|
| 500 | 46.00 ±10.51 | 49.66 ±10.37 | 48.90 ±12.68 | 49.25 ±10.95 | 45.19 ±8.20 | 46.20 ±9.26 | 48.38 8.52 | 49.41± 6.79 | 48.06± 7.09 | |
| 1000 | 6.81± 1.33 | 6.73± 1.93 | 6.30 ±0.99 | 7.07± 2.33 | 6.89 ±2.36 | 7.01± 2.21 | 7.00 2.42 | 6.67 ±1.81 | 5.86± 0.96 | |
| 2500 | 3.72± 0.57 | 4.73 ±0.47 | 4.81 ±0.49 | 4.68± 0.58 | 4.74 ±0.37 | 4.93 ±0.51 | 5.05 0.66 | 4.79 ±0.54 | 4.44± 0.61 | |
| 5000 | 2.66±3.66 | 4.01±5.2 | 4±5.4 | 4.42±6.03 | 4.48±5.58 | 4.69±6.66 | 4.61±5.9 | 4.63±6.3 | 3.9±5.5 | |
| 10000 | 1.58 ±0.33 | 3.33± 0.50 | 3.75 ±0.59 | 3.90± 0.73 | 4.06 ±0.74 | 4.30 ±0.72 | 4.21 0.80 | 4.20 ±0.77 | 3.20 ±0.65 | |

too rapidly, the offline error degrades. This happens because of the large hyper-space created by the higher dimensional search space. On the other hand, if the number of nodes in the distributed system is increased, as the number of particles is increased, the average offline error decreases rapidly. Table 5 shows the result for the same.

5 Conclusion

In this paper, we have demonstrated the use of a P2P network for optimizing DSOP. As evident, results are promising and the current setup can be utilized to find solutions for optimization problems where the number of variables or the dimensions are exceedingly large. With the current single system based optimizers, such an optimization problem will take an enormous amount of time. Cluster based systems with special purpose high bandwidth interconnect can be used to solve such large problems but creating and maintaining a cluster is not a cost effective solution for small-medium enterprises. With the use of commodity workstations, this P2P framework can be effectively used to solve such problems. However, P2P frameworks have certain problems. For example, P2P systems have trust deficit and in some cases, malicious nodes can completely mislead an optimization problem. Malicious nodes can alter true values and provide misleading gbest to the network of peer-to-peer nodes. Such malicious activity can also happen if the message communication among the set of nodes are not

secure. In our future work, we would like to concentrate on the dependability and security aspect of theP2P network from the purview of the needs of optimization problems. With such enhancements, the present system can not only work inside a single enterprise but also in the internet.

References

1. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: *ACM SIGCOMM Computer Communication Review*, vol. 31, pp. 149–160. ACM (2001)
2. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Guerraoui, R. (ed.) *Middleware 2001*. LNCS, vol. 2218, pp. 329–350. Springer, Heidelberg (2001)
3. Yang, S., Li, C.: A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments. *IEEE Transactions on Evolutionary Computation* 14(6), 959–974 (2010)
4. Li, C., Yang, S.: A clustering particle swarm optimizer for dynamic optimization. In: *IEEE Congress on Evolutionary Computation, CEC 2009*, pp. 439–446. IEEE (2009)
5. Blackwell, T., Branke, J.: Multi-swarm optimization in dynamic environments. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) *EvoWorkshops 2004*. LNCS, vol. 3005, pp. 489–500. Springer, Heidelberg (2004)
6. Nguyen, T.T., Yang, S., Branke, J.: Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation* 6, 1–24 (2012)
7. Biswas, S., Bose, D., Kundu, S.: A clustering particle based artificial bee colony algorithm for dynamic environment. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Nanda, P.K. (eds.) *SEMCCO 2012*. LNCS, vol. 7677, pp. 151–159. Springer, Heidelberg (2012)
8. Bose, D., Biswas, S., Kundu, S., Das, S.: A strategy pool adaptive artificial bee colony algorithm for dynamic environment through multi-population approach. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Nanda, P.K. (eds.) *SEMCCO 2012*. LNCS, vol. 7677, pp. 611–619. Springer, Heidelberg (2012)
9. Kundu, S., Biswas, S., Das, S., Suganthan, P.N.: Crowding-based local differential evolution with speciation-based memory archive for dynamic multimodal optimization. In: *Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference*, pp. 33–40. ACM (2013)
10. Biswas, S., Kundu, S., Das, S., Vasilakos, A.: Information sharing in bee colony for detecting multiple niches in non-stationary environments. In: *Proceeding of the Fifteenth Annual Conference Companion on Genetic and Evolutionary Computation Conference Companion*, pp. 1–2. ACM (2013)
11. Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, vol. 3. IEEE (1999)
12. Li, C., Yang, S., Nguyen, T., Yu, E., Yao, X., Jin, Y., Beyer, H., Suganthan, P.: Benchmark generator for cec 2009 competition on dynamic optimization. University of Leicester, University of Birmingham, Nanyang Technological University, Tech. Rep. (2008)
13. Tang, K., Yao, X., Suganthan, P.N., MacNish, C., Chen, Y.P., Chen, C.M., Yang, Z.: Benchmark functions for the cec2008 special session and competition on large scale global optimization. *Nature Inspired Computation and Applications Laboratory, USTC, China* (2007)

Aligned PSO for Optimization of Image Processing Methods Applied to the Face Recognition Problem

Juan Luis Fernández-Martínez¹, Ana Cernea¹, Esperanza García-Gonzalo¹, Julian Velasco¹, and Bijaya Ketan Panigrahi²

¹ Mathematics Department, Oviedo University, Spain
² Department of Electrical Engineering, IIT Delhi, India

Abstract. This paper is devoted to present the stochastic stability analysis of a novel PSO version, the aligned PSO, and its application to the face recognition problem using supervised learning techniques. Its application to the ORL database provides 100% median identification accuracy over 100 independent runs.

1 Introduction

Particle Swarm Optimization (PSO) is a bio-inspired algorithm that tries to mimic the the behavior of birds flocks or fish schooling. It was first proposed by Kennedy and Eberhart [9]. A swarm of n individuals or particles search the solution space and moves according to the equations

$$\begin{aligned} \mathbf{v}_i^{k+1} &= \omega \mathbf{v}_i^k + \phi_1^k (\mathbf{g}^k - \mathbf{x}_i^k) + \phi_2^k (\mathbf{1}_i^k - \mathbf{x}_i^k), \\ \mathbf{x}_i^{k+1} &= \mathbf{x}_i^k + \mathbf{v}_i^{k+1}, \end{aligned} \quad (1)$$

where $\omega \in \mathbb{R}$ is called the inertia weight, and ϕ_1 and ϕ_2 are random variables uniformly distributed in the intervals $[0, a_g]$ and $[0, a_l]$ with $a_g, a_l \in \mathbb{R}$. The first term in the velocity update is called the inertia damping; the second term is the social contribution and it is a stochastic discrete gradient between the positions and the global best \mathbf{g}^k , that is, the model with the lowest misfit found; finally the third term is called the cognitive contribution and it is also a stochastic discrete gradient with the best position found for each particle, $\mathbf{1}_i^k$.

This equation can be considered as a particular discretization (centered in acceleration and backward in velocity) of the continuous spring-mass system [3]:

$$\begin{cases} \mathbf{x}_i''(t) + (1 - \omega) \mathbf{x}_i'(t) + \phi \mathbf{x}_i(t) = \phi_1 \mathbf{g}(t) + \phi_2 \mathbf{1}_i(t), \\ \mathbf{x}_i(0) = \mathbf{x}_{i0}, \\ \mathbf{x}_i'(0) = \mathbf{v}_{i0}. \end{cases} \quad (2)$$

2 The Novel Aligned PSO Version

In this paper we propose to adopt the same PSO discretizations for the velocities and accelerations:

$$\begin{aligned} \mathbf{x}_i''(k) &\simeq \mathbf{x}_i^{k+1} - 2\mathbf{x}_i^k + \mathbf{x}_i^{k-1}, \\ \mathbf{x}_i'(k) &\simeq \mathbf{x}_i^k - \mathbf{x}_i^{k-1}, \end{aligned} \quad (3)$$

and to approximate the position by an average of \mathbf{x}_i^k and \mathbf{x}_i^{k-1} as follows:

$$\frac{\mathbf{x}_i^k + \mathbf{x}_i^{k-1}}{2}.$$

This approximation is commonly used in convection-diffusion problems.

Then, the new PSO algorithm, named aligned PSO, becomes:

$$\begin{aligned} \mathbf{v}_i^{k+1} &= \omega \mathbf{v}_i^k + \frac{\phi_1^k}{2}(\mathbf{g}^k - \mathbf{x}_i^k) + \frac{\phi_2^k}{2}(\mathbf{l}_i^k - \mathbf{x}_i^k) + \\ &\quad + \frac{\phi_1^k}{2}(\mathbf{g}^{k-1} - \mathbf{x}_i^{k-1}) + \frac{\phi_2^k}{2}(\mathbf{l}_i^{k-1} - \mathbf{x}_i^{k-1}), \\ \mathbf{x}_i^{k+1} &= \mathbf{x}_i^k + \mathbf{v}_i^{k+1}. \end{aligned} \tag{4}$$

It can be observed that a new term, involving the attractors \mathbf{g}^{k-1} and \mathbf{l}_i^{k-1} , appears in the velocity update. This fact affects the center of attraction for each particle, that now becomes:

$$\mathbf{o}_i^k = \frac{\phi_1^k (\mathbf{g}^k + \mathbf{g}^{k-1}) + \phi_2^k (\mathbf{l}_i^k + \mathbf{l}_i^{k-1})}{2 (\phi_1^k + \phi_2^k)},$$

instead of:

$$\mathbf{o}_i^k = \frac{\phi_1^k \mathbf{g}^k + \phi_2^k \mathbf{l}_i^k}{(\phi_1^k + \phi_2^k)},$$

in the case of the standard PSO version (1)

2.1 The First Order Stability Region

In this section we briefly analyze the stochastic stability of the aligned PSO version, following the methodology shown in [3,4,5]. The first order dynamical system associated with the algorithm (4) is:

$$\begin{pmatrix} E(\mathbf{x}_i^{k+1}) \\ E(\mathbf{x}_i^k) \end{pmatrix} = A_\mu \begin{pmatrix} E(\mathbf{x}_i^k) \\ E(\mathbf{x}_i^{k-1}) \end{pmatrix} + c_\mu, \tag{5}$$

where

$$A_\mu = \begin{pmatrix} E(A) & E(B) \\ 1 & 0 \end{pmatrix},$$

and

$$\begin{aligned} A &= \frac{2\omega - \phi^k + 2}{2}, \\ B &= -\frac{\phi^k + 2\omega}{2}. \end{aligned} \tag{6}$$

The iteration matrix A_μ depends on ω and $\bar{\phi} = E(\phi)$. System (5) is stable when the spectral radius of A_μ is less than one. The first order stability region turns out to be the triangular region given by:

$$S_1 = \{(\omega, \bar{\phi}) : -1 < \omega < 1, 0 < \bar{\phi} < 2(1 - \omega)\}. \tag{7}$$

Also the parabola

$$\bar{\phi} = 6 - 4\sqrt{2 + 2\omega} + 2\omega, \tag{8}$$

marks the limit between the real and complex eigenvalues of A_μ . Figure 1a shows the isolines of the first order spectral radius.

2.2 The Second Order Stability Region

The second order stability dynamical system associated with (4) is:

$$\begin{pmatrix} E\left(\left(\mathbf{x}_i^{k+1}\right)^2\right) \\ E\left(\mathbf{x}_i^{k+1}\mathbf{x}_i^k\right) \\ E\left(\left(\mathbf{x}_i^k\right)^2\right) \end{pmatrix} = A_\sigma \begin{pmatrix} E\left(\left(\mathbf{x}_i^k\right)^2\right) \\ E\left(\mathbf{x}_i^k\mathbf{x}_i^{k-1}\right) \\ E\left(\left(\mathbf{x}_i^{k-1}\right)^2\right) \end{pmatrix} + c_\sigma, \tag{9}$$

with

$$A_\sigma = \begin{pmatrix} E(A^2) & 2E(AB) & E(B^2) \\ E(A) & E(B) & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

where matrix A_σ depends on ω and $\bar{\phi} = E(\phi)$ and $E(\phi^2)$. Taking into account that the distributions of ϕ_1 and ϕ_2 are uniform and naming $\beta = a_g/a_l$, we have

$$\bar{\phi} = \bar{\phi}_1 + \bar{\phi}_2 = \frac{a_g}{2} + \frac{a_l}{2} = (1 + \beta) \frac{a_g}{2}, \tag{10}$$

$$\text{Var}(\phi) = \frac{a_g^2}{12} + \frac{a_l^2}{12} = \frac{a_g^2}{12} (1 + \beta^2) = \frac{1 + \beta^2}{3(1 + \beta)^2} \bar{\phi}^2,$$

$$E(\phi^2) = \text{Var}(\phi) + \bar{\phi}^2 = \frac{2(2 + 3\beta + 2\beta^2)}{3(1 + \beta)^2} \bar{\phi}^2.$$

Now the iteration matrix A_σ depends on ω , $\bar{\phi}$ and β . The second order system (9) is stable in region

$$S_2 = \{(\omega, \bar{\phi}, \beta) : -1 < \omega < 1, 0 < \bar{\phi} < m_\beta(1 - \omega)\}, \tag{11}$$

where

$$m_\beta = \frac{3(1 + \beta)^2}{2 + 3\beta + 2\beta^2},$$

is the slope of the second order upper stability limit. This region is a triangle embedded in S_1 . Figure 1b also shows the second order spectral radius isolines. The upper stability limit intersects the ω axis in $\omega = 1$. Its slope is maximum when $\beta = 1$, being its value $m_\beta = 12/7$. This slope is minimum when $\beta \rightarrow 0$ and when $\beta \rightarrow \infty$ being its value $m_\beta = 3/2$. The geometry of the first and second order stability regions for the aligned PSO is much simpler than the ones corresponding to the standard PSO version.

2.3 Numerical Experiments

We have performed several numerical experiments using different benchmark functions currently used in practice. We had calculated the median logarithmic error over 50 simulations for a grid of $(\omega, \bar{\phi})$ points covering the first order stability region. We have used 20, 40 and 100 particles for dimensions 10, 30 and

50 respectively. We have used as objective functions the Rosenbrock, Griewank, Rastrigin and Sphere benchmark functions. Figure 2 shows the results that have been obtained for 30 dimensions with the Rosenbrock and Griewank functions with the PSO and aPSO algorithms. It can be observed that the low misfit $(\omega, \bar{\phi})$ points are located in a boomerang type zone that includes the upper border of the aPSO second stability region. In the aPSO case the boomerang part tend to vanish when the dimension of the search space increases. Also, PSO and aPSO provide similar minimum misfits.

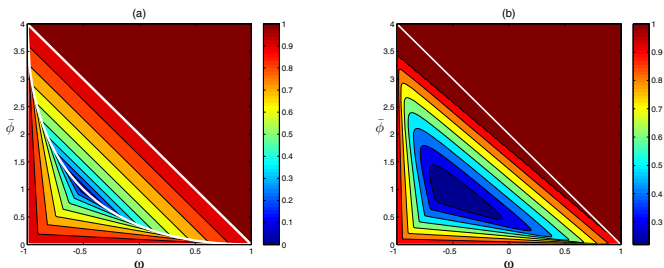


Fig. 1. Spectral radius isolines for (a) first and (b) second order stability regions for $\beta = 1$

3 Application to the Face Recognition Problem

3.1 Image Classification

The automatic image recognition problem consists of classifying a given probe image I providing a database of training images. Mathematically the problem can be formulated as follows: given a database of training images

$$B_d = \{I_k \in S_{(n,m)}(\mathbb{N}) : k = 1, \dots, N\}, \tag{12}$$

characterized by a set of labels

$$C_d = \{C_k \in \{1, 2, \dots, q\}, k = 1, \dots, N\}, \tag{13}$$

and a new incoming image $I \notin B_d$, the problem consists in estimating its class C_I^* . In this definition $S_{(n,m)}$ is the space gray-scale images of size $m \times n$. In this problem the learning database typically contains N_p poses of each of the q individual classes, that is $N = q \cdot N_p$. To perform the classification it is necessary to construct a learning algorithm for the class prediction:

$$C^* : S_{(n,m)} \rightarrow C_d. \tag{14}$$

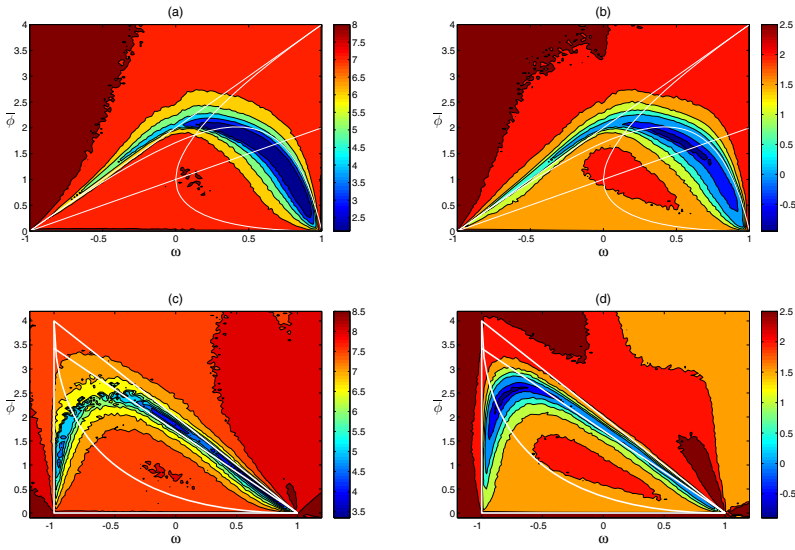


Fig. 2. Logarithmic median error in 30 dimensions for (a) PSO with Rosenbrock function, (b) PSO with Griewank function, (c) aPSO with Rosenbrock function and (d) aPSO with Griewank function

In the case of non-supervised learning the methodology is as follows:

1. First finding the image $I_k \in B_d$ such as:

$$d(I, I_k) = \min_{j \in B_d} d(I, I_j), \tag{15}$$

where d is a suitable distance (or norm) criterium defined over $S_{(n,m)}$.

2. Once this image has been found: $C_I^* = C_{I_k} = C_k$.

The images are represented by a feature vector calculated for each individual method of analysis (or attribute). Naming $\mathbf{v}_i^k \in \mathbb{R}^{s_k}$ the feature vector of image I_i according to the attribute k , the distance between two images I_i and I_j is defined as follows:

$$d(I_i - I_j) = \|\mathbf{v}_i^k - \mathbf{v}_j^k\|_p, \tag{16}$$

where p is a certain norm defined over the k -attribute space (\mathbb{R}^{s_k}). The success of the non-supervised classification implicitly depends on the relationship between the k -attribute, the adopted criteria (norm or cosine) and the class information. Not all the attributes used in this paper perform equally. Attributes are described in section 3.2.

In the case of supervised classification the learning method C^* depends also on a set of parameters \mathbf{m} that has to be tuned using class information coming from a subset of images of the database (testing database). The algorithm of supervised classification of a test images I used in this paper is as follows:

1. From every individual non-supervised classifier built using the different attributes, we retain the first N_{first} images that are closer to I .
2. Based on this classification a matrix $M \in M_{N_{first} \times N_a}$ is built, containing the N_{first} image candidates for each of the N_a attributes ($N_a = 5$).
3. The array $\mathbf{c} \in \mathbb{R}^{N_d}$ containing the N_d different candidates is formed. For any image candidate $I_c \in \mathbf{c}$, its score according to the set of positions (i, j) in matrix M is calculated as follows:

$$s(I(i, j)) = (N_{first} - j + 1) \cdot \mathbf{w}(i),$$

where \mathbf{w} is a vector of weights corresponding to the trust factors assigned to any individual classifier (attribute).

4. After calculating the scores for all images in matrix M , the final classification of the test image I is performed by selecting the image with the major score among all the candidates of M matrix.

The aligned PSO version is used to optimize the parameter N_{first} and the set of weights \mathbf{w} . For that purpose the learning database will be divided in two different parts, one for learning and the other for optimizing. The results of the supervised classifiers will be obtained with a validation database of images that were not used in the supervised learning process.

3.2 Image Attributes

In this paper we have used the following list of attributes, statistical based (histogram and variogram), spectral (discrete cosine transform and discrete wavelet transform), and image segmentation/regional descriptors (texture). All these attributes can be calculated for gray scale and color images, both, locally or globally. In the case of global analysis the attribute features are calculated over the whole size of the image, meanwhile in the case of local features, the image is divided into blocks. For each block the local attributes are computed and the final feature vector is formed by merging all the local attributes into an unique vector, always computed in the same order. In this paper we have used a partition of the images into 8×4 blocks, nevertheless finer subdivisions could be also adopted.

Color Histograms. An image histogram describes the frequency of the brightness in the image. The shape of the histogram provides information about the nature of the image [11].

For a gray-scale digital image I the histogram represents the discrete probability distribution of the gray-levels in the image. For this purpose the gray-scale space $[0, 255]$ for an 8-bit image) is divided into L bins, and the number of pixels in each class n_i , ($i = 1, L$) is calculated. In this case the attribute vector has dimension L :

$$H_I = (n_1, \dots, n_L).$$

Relative frequencies can be also used by dividing the absolute frequencies n_i by the total number of pixels in the image.

In the case of *RGB* images the histogram is calculated for each color channel I_R , I_G and I_B , and then all the channels histograms are merged together, as follows:

$$H_I = (H(I_R), H(I_G), H(I_B)).$$

Variogram. The variogram of an image describes the spatial distribution in each color channel. In spatial statistics [6] the variogram describes the degree of spatial dependence of a spatial random field or stochastic process, the gray-scale in this case. For a given value of vector h , defined by a modulus and direction, the variogram is an index of dissimilarity between all pairs of values separated by vector h .

The omnidirectional p -variogram is the mean of the p -absolute difference between the color values of the $N(h)$ pairs of pixels that are located at the same distance h :

$$\gamma_i(h) = \frac{1}{N(h)} \sum_{k=1}^{N(h)} |c_i(x_k) - c_i(x_k + h)|^p. \tag{17}$$

Usually $p = 2$. To compute the variogram each color channel (matrix) is transformed into the corresponding color vector $c_i(x)$. Typically $N(h)$ is limited to one third of the total number of pixels. The number of classes that have been considered in this case was $N(h) = 100$.

Texture Analysis. Texture analysis of an image consists in analyzing regular repetitions of a pattern [10]. In this paper, we use the spatial gray level co-occurrence matrix to describe an image texture.

The gray level co-occurrence matrix (GLCM), or spatial dependence matrix of an image I is an estimate of the second-order joint probability function $P_{d,\theta}(i, j)$ of the intensity values of two pixels i and j located at a distance d apart (measured in number of pixels) along a given direction θ [2,1]. Typically the GLCM is calculated for different pairs of d and θ . Figure 3 shows the spatial relationships between a pixel and its adjacent pixels, and the corresponding displacement vector (d, θ) . Different statistical moments can be calculated from the GLCM ma-

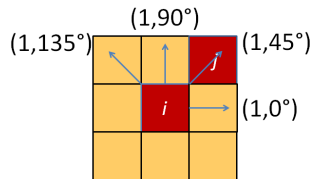


Fig. 3. Spatial relationships of a pixel i with its adjacent pixels

trix, such as contrast, homogeneity, squared energy, correlation and entropy [2]. In the present case we have used a lag $d = 1$ for the directions $0^\circ, 45^\circ, 90^\circ, 135^\circ$. This analysis provides an attribute vector of dimension 20 for each image.

Discrete Cosine Transform (DCT). DCT is a free-covariance model reduction technique that attempts to decorrelate 2D images by projecting the rows and columns of the incoming image into cosines of increasing frequency. DCT has been used by Hafeed and Levine [7] in face recognition, showing that DCT applied to normalized images is very robust to variations in geometry and lightning.

Mathematically, DCT is a discrete Fourier transform that expresses a signal in terms of a sum of sinusoids with different frequencies and amplitudes. For an image I_k the DCT is defined as follows:

$$D(u, v) = c(u)c(v) \sum_{i=0}^{s-1} \sum_{j=0}^{n-1} D_{(i,j)}$$

where

$$D_{(i,j)} = I_k(i, j) \cdot \cos \frac{\pi(2i + 1)u}{2s} \cos \frac{\pi(2j + 1)v}{2n},$$

$u = 0, \dots, s - 1$, and $v = 0, \dots, n - 1$, being

$$c(\alpha) = \begin{cases} \frac{1}{\sqrt{N}}, & \text{if } \alpha = 0, \\ \sqrt{\frac{2}{N}}, & \text{if } \alpha \neq 0. \end{cases}$$

N is either the number of rows (s) or columns (n) of the image. The DCT can be expressed in matrix form as an orthogonal transformation

$$D_{CT} = U_{DC} I_k V_{DC}^T,$$

where matrices U_{DC} and V_{DC} are orthogonal. This transformation is separable and can be defined in higher dimensions. The feature vector of an image I_k is constituted by the $q_1 - q_2$ block of D_{CT} , $D_{CT}(1 : q_1, 1 : q_2)$, where q_1, q_2 are determined by energy reconstruction considerations using the Frobenius norm of the image I_k .

Discrete Wavelet Transform (DWT). Wavelets are compact functions (defined over a finite interval) with zero mean and some regularity conditions (vanishing moments). The Wavelet transform converts a function into a linear combination of basic functions, called wavelets, obtained from a prototype wavelet through dilatations, contractions and translations. DWT was applied to face recognition by Kakarwal and Deshmukh [8].

The discrete wavelet transform (D_{WT}) of an image $I \in \mathcal{M}(m, n)$ is defined as follows:

$$D_{WT} = U_W^T I V_W,$$

where U_w and V_w are two orthogonal matrices constructed as follows:

$$U_W = \begin{bmatrix} H \\ G \end{bmatrix}_m^T, V_W = \begin{bmatrix} H \\ G \end{bmatrix}_n^T,$$

where H represents a low pass or averaging portion of the wavelet filter, and G is the high pass or differencing portion. In the present case we have used the transform having a maximum number of vanishing moments: the Daubechies-2 family.

4 Numerical Results

To perform the numerical analysis we have used the ORL database of faces provided by AT&T Laboratories Cambridge. The ORL database contains 400 grey scale images, ten different poses of 40 distinct individuals taken during a period of two years. All the images were taken against a dark homogeneous background, varying the lighting, facial expressions and facial details. The database provides upright and frontal poses. The size of each image is 92x112 pixels, with 256 grey levels per pixel. In all the experiments over ORL, the learning database is composed of five poses of each individual, that are randomly selected. In the supervised learning procedure three of the poses are used to learn and the other two are used to optimize the learning parameters. The rest of the poses in the database are used as probe images for establishing the accuracy of the classification for each spectral technique, using both, global and local features. For each attribute the classification is performed 100 different times, randomly choosing the learning database and the set of probe images (200 images). Table 1 shows the median accuracy obtained for each of the individual attributes. The higher accuracies are obtained for the local histogram (98%), the DWT (95.5%) and the DCT (95.25%). The variogram and the texture analysis provides lower median accuracies. Using the supervised learning technique explained above a median accuracy of 100% is obtained. Table 2 shows the parameters provided by the aligned PSO, to perform the classification.

Table 1. Accuracies of each individual attribute

| Classifier (Attribute) | Histogram | Variogram | Texture | DWT | DCT |
|------------------------|-----------|-----------|---------|-------|-------|
| Median Accuracy | 98.00 | 90.00 | 90.75 | 95.50 | 95.25 |

Table 2. Parameters for supervised learning algorithm optimized by aPSO

| Attribute | Histogram | Variogram | Texture | DWT | DCT | Nfirst |
|-----------|-----------|-----------|---------|-------|-------|--------|
| Weights | 65.21 | 61.84 | 80.15 | 80.02 | 62.14 | 6.00 |

5 Conclusions

In this paper we present a novel PSO version, the aligned PSO that is deduced from the PSO continuous model adopting for the trajectories a discretization that it is in accord to the velocity discretization terms. This approach is currently used in diffusion-convection problems. First and second order stochastic stability analysis is also performed following the methodology that was used in the past for other PSO family members. Numerical experiments with benchmark functions show that performing aligned PSO parameter sets of inertia and global and local accelerations are located close to the upper limit of the second order stability region where exploration capabilities are very high. Finally the aligned PSO is applied to optimize the parameters of a supervised learning algorithm to solve the face recognition problem. Its application to the ORL database provides 100% median identification accuracy over 100 independent runs.

References

1. Aksoy, S., Haralick, R.M.: Content-based image database retrieval using variances of gray level spatial dependencies. In: Ip, H.H.-S., Smeulders, A.W.M. (eds.) *MILNAR 1998*. LNCS, vol. 1464, pp. 3–19. Springer, Heidelberg (1998)
2. Bharati, M.H., Jay Liu, J., MacGregor, J.F.: Image texture analysis: methods and comparisons. *Chemometrics and Intelligent Laboratory Systems* 72(1), 57–71 (2004)
3. Fernández-Martínez, J.L., García-Gonzalo, E.: The generalized PSO: a new door to PSO evolution. *J. of Artif. Evol. and Appl.* 2008, 1–15 (2008)
4. Fernández-Martínez, J.L., García-Gonzalo, E.: The PSO family: deduction, stochastic analysis and comparison. *Swarm Intell.* 3(4), 245–273 (2009)
5. Fernández-Martínez, J.L., García-Gonzalo, E.: Stochastic stability analysis of the linear continuous and discrete PSO models. *IEEE Trans. Evol. Comput.* 15(3), 405–423 (2011)
6. Goovaerts, P.: *Geostatistics for natural resources evaluation*. Applied geostatistics series. Oxford University Press (1997) (Incorporated)
7. Hafed, Z.M., Levine, M.D.: Face recognition using the discrete cosine transform. *Int. J. Comput. Vision* 43(3), 167–188 (2001)
8. Kakarwal, S., Deshmukh, R.: Wavelet transform based feature extraction for face recognition. *International Journal of Computer Science and Application Issue 1*, 100–104 (2010)
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings IEEE International Conference on Neural Networks (ICNN 1995)*, Perth, WA, Australia, vol. 4, pp. 1942–1948 (November–December 1995)
10. Srinivasan, G.N., Shobha, G.: Statistical texture analysis. In: *Engineering And Technology Proceedings of World Academy of Science*, vol. 36, pp. 1264–12699 (2008)
11. Umbaugh, S.E.: *Computer Vision and Image Processing: A Practical Approach Using CVIPtools*. Prentice Hall Professional Technical Reference (1998)

Optimal Operation Management of Transmission System with Fuel Cell Power Plant Using PSO

S. Vidyasagar, K. Vijayakumar, and D. Sattianadan

Department of EEE, SRM University,
Chennai -603203, Tamil Nadu, India

{mailsvs,sattia.nadan}@gmail.com, kvijay_srm@rediffmail.com

Abstract. In this paper Particle Swarm Optimization (PSO) algorithm to solve the Optimal Operation Management (OOM) of transmission system is presented. The purpose of the OOM problem is to decrease the total electrical energy cost, real power losses in transmission system and the total pollutant emission produced by fuel cells. Fuel cell power plants (FCPPs) have been taken into a great deal of consideration in recent years. The continuing growth of the power demand together with environmental constraints is the great interest to use FCPPs in power system. One of the most important issues in the power system is optimal operation management which can be effected by FCPPs. The various objectives of OOM problem is solved as single and /or multi – objective problem using PSO technique. Numerical results on IEEE 30 bus test system have been presented.

Keywords: Fuel Cell Power Plant(FCPP), Particle Swarm Optimization(PSO), Optimal Operation Management(OOM).

1 Introduction

Power restructuring have created increasing interest in the wide use of distributed generation (DG) in the electric power system. A wide variety of DG technologies and types are reported: renewable energy sources such as wind generators, photovoltaic cells, micro-turbines, fuel cells, and energy storage devices such as batteries, etc. [1,2]. The use of fuel cells (FCs)are preferred because with low power generation a high efficiency can be obtained FCs appears as one of the most promising solution due to their good efficiency even at partial load, and especially due to their clean electric generation, with only water and heat as by-products. Also, their low noise and static operation makes them suitable to be used even in domestic generations [3, 4]. Studies carried out by researchers have revealed that fuel cell power plants (FCPPs) contribution in energy production systems will be enhanced more than 25% in near future [5]. An optimal operation management with regard to FCPPs in distribution system is discussed in [6]. The impact of FCPPs on the power transmission system-shas not been discussed in literatures. That is the motive of choosing FCPPs in operational management in transmission system. The objective functions include: the total electrical energy cost, total electrical energy losses and the total pollutant emission by the fuel cells. These objectives are optimized as single and / or multi objective

optimization problem by finding optimal location and rating of FCCPP's, Static capacitors and transformer tap ratio. Due to the simple, easy implementation and quick convergence PSO has attracted much attention and has obtained wide applications in various kinds of nonlinear optimization problems. The multi-objective PSO based algorithm for optimal operation of grid is discussed in [7]. The efficient constraint handling algorithm is given in [8]. This is the motive to choose PSO for solving the operation management problem in this paper.

The paper is organized in the following manner: In section -2, the Problem formulation is given, In section-3, the modelling of FCCPP is presented, In section -4 the particle swarm optimization algorithm for OOM is presented. In section 5 the results & discussion are given and in section 6 the conclusions are arrived.

2 Problem Formulation

In the Optimal Operation Management (OOM) problem the various objectives considered are minimization of total electrical energy cost, total electrical energy losses and the total emission by FCCPP's subject of various constraints. These objectives are solved using PSO as single and / or multi - objective optimization problem. The various cases considered in this study are:

- Case (i) Minimization of total cost of electrical energy
- Case (ii) Minimization of total emission of FCCPP'S
- Case (iii) Minimization of real power losses and total cost of energy.
- Case (iv) Minimization of emission and power losses
- Case (v) Minimization of cost of energy, power loss and emission all together.

2.1 Minimization of the Total Cost of Electrical Energy Generated by FCCPPs

The total cost of energy is the sum of cost of electrical energy generated by FCCPP's and the cost of active power generation of generators. So the objective is to minimize total cost of energy.

$$\text{Min } f_1(X) = \text{Total Cost} = C_{FC} + \sum_{n=1}^{NG} C_n \tag{1}$$

Where, C_{FC} is the cost of electrical energy generated by FCCPPs and is given as follows

$$C_{FC} = 0.04 \text{ \$/kwh} \times \sum_{j=1}^{Ng} P_{gj} \tag{2}$$

Where P_{gj} is the real power generated by FCCPPs and

$$C_n = A(P_{gi})^2 + B(P_{gi}) + C \tag{3}$$

Where A, B and C are the cost coefficient of the generators and P_{gi} is the active power generation by the generators.

2.2 Minimization of the Total Emissions

The emission of FCPP's depends on the quantity of power generated by FCPP's, If the power produced is more, the cost and the emission will be reduced. So the objective is to minimize the total cost of energy as well as emission of FCPP's. It is given as follows

$$\text{Min}f_2(X) = \text{Emission} = E_{FC} + C_{FC} + \sum_{n=1}^{NG} C_n \quad (4)$$

Where, E_{FC} is the emission of FCPP, E_{FC} can be defined as .

$$E_{FC} = NO_{xFC} + SO_{2FC} = (0.01361 + 0.00272)kg/Mwh \times \sum_{j=1}^{Ng} P_{gj} \quad (5)$$

Where NO_{xFC} is the nitrogen oxide pollutant of FCPP and SO_{2FC} is the sulphur oxide pollutants of FCPP. The control variables for this case are the real power produced by FCPP's and the active power generation of the generator in the system.

2.3 Minimization of the Power Losses

Usually FCPP's are installed in the distribution networks. So its installation leads to reduce the power losses in the system .Larger the quantity of power generated by FCPP's, lesser will be the power loss but the cost will also be higher. So the objective is to minimize the loss as well as the cost as given below.

Minimizing the electrical energy losses of the distribution network in the presence of FCPPs is of great importance in optimal operation problem. So the objective considered is as follows.

$$\text{Min}f_3(X) = C_{FC} + P_{Loss} + \sum_{n=1}^{NG} C_n \quad (6)$$

$$P_{loss} = \sum_{i=1}^{Ng} R_i |I_i^2|$$

x is the state vector which includes active power of FCPPs(\bar{P}_G), Tap of transformers

(\bar{T}_{ap}) and capacitor reactive power(\bar{Q}_c), that can be described as follows:

$$x = [\bar{P}_G, \bar{T}_{ap}, \bar{Q}_c]$$

$$\begin{aligned} \bar{P}_G &= [\bar{P}_{g1}, \bar{P}_{g2} \dots \dots, \bar{P}_{gNg}]; & N_g \text{ is the number of FCPPs} \\ \bar{T}_{ap} &= [\bar{T}_{ap1}, \bar{T}_{ap2} \dots \dots, \bar{T}_{apNt}]; & N_t \text{ is the number of transformers} \\ \bar{Q}_c &= [\bar{Q}_{c1}, \bar{Q}_{c2} \dots \dots, \bar{Q}_{cNc}]; & N_c \text{ is the number of capacitors} \end{aligned}$$

2.4 Minimization of the Emission and Power Losses

Using FCPP's the losses will be minimized but the pollutant emission will be there. Use of large capacity of FCPP's leads to less loss and more emission, so the objectives are to minimize the losses and emission together with cost of generators.

$$\text{Min}f_4(X) = E_{FC} + P_{Loss} + \sum_{n=1}^{NG} C_n \quad (7)$$

The control variables are same as given in section 2.3

2.5 Minimization of Emission, Cost and Power Losses

Here the objective is to minimize the emission produced by fuel cells , cost of electrical energy generated by fuel cells , generation cost of generators and total power losses together.

$$\text{Min}f_5(X) = E_{FC} + C_{FC} + P_{Loss} + \sum_{n=1}^{NG} C_n \quad (8)$$

The various constraints considered for all the problems mentioned earlier are mentioned below

➤ *Power balance constraints*

$$P_{gi} - P_{di} = \sum_{j=1}^{NB} |V_i| |V_j| |Y_{ij}| \cos(\delta_i - \delta_j - \theta_{ij}) \quad (9)$$

$$Q_{gi} - Q_{di} = \sum_{j=1}^{NB} |V_i| |V_j| |Y_{ij}| \sin(\delta_i - \delta_j - \theta_{ij}) \quad (10)$$

➤ *Active & Reactive Power Constraints of Generators:*

$$P_{gi}^{\min} \leq P_{gi} \leq P_{gi}^{\max}; i = 1, 2, \dots, N_g \quad (11)$$

$$Q_{gi}^{\min} \leq Q_{gi} \leq Q_{gi}^{\max}; i = 1, 2, \dots, N_g \quad (12)$$

➤ *Active Power Constraints of FCPPs:*

$$P_{min,FC}^t \leq P_{min,FC}^t \leq P_{max,FC}^t \leq P_{gi}^t \leq P_{max,FC}^t \quad (13)$$

$P_{min,FC}^t$ is minimum active power of the i^{th} FCPP and $P_{max,FC}^t$ is maximum active power of the i^{th} FCPP.

➤ *Transmission Line Limits:*

$$|P_{ij}^{\text{line}}|^t < P_{ij,max}^{\text{line}} \quad (14)$$

$|P_{ij}^{line}|^t$ and $P_{ij,max}^{line}$ are the active power flowing in lines and active power limit of line connected between the nodes i and j, respectively.

➤ *Bus Voltage Magnitude:*

$$V_{min} \leq V_i^t \leq V_{max} \quad (15)$$

V_i^t , V_{max} and V_{min} are the voltage magnitudes of the i^{th} bus, the maximum and minimum values of voltage magnitude, respectively.

➤ *Tap of Transformers:*

$$T_{api}^{min} < T_{api}^t < T_{api}^{max} \quad (16)$$

T_{api}^{min} and T_{api}^{max} are the minimum and maximum tap positions of the i^{th} transformer respectively.

3 Fuel Cell Power Plant Modelling

The fuel cell is one of the developments in alternate energy field, In simple word, it is an electrochemical energy generating device. It has become one of the popular and interesting aspects of modern technology. There are a lot of things that are yet to be developed in this field and also fuel cell technology is vast and involves various applications.

Because of their cleanness, good efficiency and high reliability, there are different kinds of FCs according to their characteristics.

Generally, FCPP's can be modelled either as PV or PQ models. It must be taken into account that when FCPP's are considered as the PV models, if they generate reactive power to maintain their voltage magnitudes. In this paper, the FCPPs are modelled as the PQ model with simultaneous three-phase control.

4 The Particle Swarm Optimization (PSO) Algorithm

PSO method is a population-based optimization technique that was first introduced by Kennedy and Eberhart [9] in which each solution called "particle" flies around a multidimensional search space. During the flight, every particle adjusts its position according to its own experience, as well as the experience of neighbouring particles, using the best position encountered with itself and its neighbours. The swarm direction of a particle is defined by its history experience and the experience of its neighbours. A particle status on the search space is described by two factors: its position and velocity, which are updated by following equations:

$$V_i^{k+1} = V_i^k + C_1 \times rand_1 \times (pbest_i - x_i^k) + C_2 \times rand_2 \times (gbest_i - x_i^k) \quad (17)$$

$$x_i^{k+1} = x_i^k + V_i^{k+1} \quad (18)$$

Where k is the current iteration number and C_1 and C_2 are weighting factors of the stochastic acceleration terms, which pull each particle towards the P_{best} and G_{best} positions, $rand1$ and $rand2$ are two random functions in the range of $[0,1]$, x_i^k is the best previous experience of i^{th} particle that is recorded and P_{best} is the best particle among the entire population. The Equation (18) is used for the calculation of i^{th} particle's velocity considering three terms: the particle's previous velocity, the distance between the particle's best previous and current positions, and finally, the distance between the position of the best particle in the swarm and i^{th} particle's current position.

4.1 Application of PSO Algorithm to OOM Problem

To apply the PSO algorithm in the MOOM problem, the following steps are carried out.

- Step 1. Read the input data: Input data includes system parameters, line impedance, characteristics of FCPPs, emission functions and prices of Fuel cells, characteristics of generators, limits of generator, etc.
- Step 2. Set the generation count $t=0$.
- Step 3. Initialize PSO parameters: the size of population, initial velocity constants $C1$ & $C2$.
- Step 4. Generate particles (refer to a set of control variables) within their limits.
- Step 5. Run load flow and find the power flow in the lines.
- Step 6. Calculate the fitness values of each particle of population using objective function
- Step 7. Find g_{best} and p_{best} based on the fitness values and update position and velocity of each particle using equations. Eq(18)
- Step 8. Check for convergence, if converged stop, else, go to step (5)

5 Results and Discussion

The proposed PSO algorithm is tested on IEEE 30 bus test system. This system has 6 generators, 3 fuel cells, 21 loads and 41 transmission lines with total load of 283.4 MW and 126.2 MVAR. Matlab coding has been done for different cases: Base case and contingency case.

Base Case:

Load flow analysis is done for IEEE 30 bus system using Newton Raphson's Method without inclusion of FCPPs and it is found that the power flows in all the lines are within the limits and generation is equal to demand and losses. The generation cost & losses are minimized using PSO method by varying the generation of 6 generators within the limits. The generator real power, the cost of generator & losses for base case is given in Table 1.

Table 1. Results for active power generation cost& using PSO for base case

| Generators | Generation (MW) | Cost of generator (\$/hr) | Losses (MW) |
|------------|-----------------|---------------------------|-------------|
| 1 | 126.75 | 933.64 | 14.47 |
| 2 | 84.11 | | |
| 5 | 30.59 | | |
| 8 | 18.12 | | |
| 11 | 12.11 | | |
| 13 | 29.10 | | |

Contingency Case: 120% Load Demand

An increase in load by 20% leads to power flow limit violation in line 1-2 and the generation is not able to meet the demand. Actual Power flow limit in line 1-2 is 100 MW. When the load increased by 120% the Power flow in line 1-2 is 130.82 MW. So the optimum location, rating of FCCPs & generators are obtained using PSO.

CASE 1: The total cost of electrical energy is minimized by varying the generation of 6 participating generators, the rating and location of FCPPs, when the load is increased by 120%. The power flow in the line 1-2 is 34.75 MW. It is observed that by installing FCPP's with proper rating and location, the line over load is relieved and also the cost of the system is reduced using PSO. It is found that the power flow in the line 1-2 is 34.75 MW which is well within the limit .The results obtained are shown in Table 2.

Table 2. Results for minimization total cost of electrical energy

| Location of FCPP | Rating of FCPP (kW) | Rating of Generators (MW) | Cost of FCPPs (\$/hr) | Cost of Generators (\$/hr) | Total cost (\$/hr) |
|------------------|-------------------------|---------------------------|-----------------------|----------------------------|--------------------|
| 29 17 4 | 22.61 27.92 33.56 | 141.45 | 3363.6 | 966.19 | 4329.79 |
| | | 46.02 | | | |
| | | 29.62 | | | |
| | | 18.06 | | | |
| | | 12.20 | | | |
| | | 12.28 | | | |

CASE 2: The emission produced by FCPPs & the total cost of electrical energy is minimized by varying generation of generators, the rating and location of FCPPs for contingency case. Power flow in the line 1-2 is 61.97 MW. It is found that with the use of FCPPs in proper place and rating, the line loading is relieved and also the emission is optimized with reduced cost. The results obtained are given in table 3.

Table 3. Results for minimization of emission produced by FCPP

| Location of FCPP | Rating of FCPP (KW) | Rating of Generators (MW) | Emission (Kg/hr) | Cost of Generators (\$/hr) | Cost of FCPP's (\$/hr) | Total Cost (\$/hr) |
|------------------|---------------------|---------------------------|------------------|----------------------------|------------------------|--------------------|
| 22 | 5.87 | 134.50 | 1.417 | 1001.8 | 3472 | 4473.8 |
| 24 | 48.05 | 39.12 | | | | |
| 20 | 32.88 | 29.84 | | | | |
| | | 32.69 | | | | |
| | | 21.71 | | | | |
| | | 19.34 | | | | |

CASE 3: The cost of electrical energy generated by FCPPs and the total electrical energy losses is minimized by varying the generation of generators, the rating and location of FCPPs, capacitive reactive power and tap of transformers when the load is increased by 120%. The Power flow in the line 1-2 is 53.27 MW. In this case the cost & loss are less, when compared to case 1 and case 2 but the cost of FCPP's is slightly high and is shown in Table 4.

Table 4. Results for minimization of cost and power losses

| Location of FCPP | Rating of FCPP (KW) | Rating of Generators (MW) | Tap position of transformer | Capacitive reactive power (KVAR) | Losses (Kw) | Cost of FCPPs (\$/hr) | Cost of Generators (\$/hr) | Total cost (\$/hr) |
|------------------|---------------------|---------------------------|-----------------------------|----------------------------------|-------------|-----------------------|----------------------------|--------------------|
| 19 | 12.79 | 95.52 | A(11)=0.95 | Qsh(11)= | 10.72 | 2425.6 | 958.84 | 3384.84 |
| 20 | 13.89 | 70.69 | A(12)=0.98 | 18.35 | | | | |
| 23 | 33.96 | 43.57 | A(15)=0.94 | Qsh(24)= | | | | |
| | | 31.15 | A(36)=0.91 | 11.96 | | | | |
| | | 17.40 | | | | | | |
| | | 22.73 | | | | | | |

CASE 4: The emission produced by FCPPs and the total electrical energy losses are minimized by varying generation of 6 generators, the rating and location of FCPPs, capacitive reactive power and tap of transformers when the load is increased by 120%. The power flow in the line 1-2 is 20.90 MW. In this case the results obtained are cost of generators, emission and losses. The cost of generator and losses are less whereas emission is slightly higher when compared to earlier cases as given in Table 5.

Table 5. Results for minimization of emission and power losses

| Location of FCPP | Rating of FCPP (KW) | Rating of Generators (MW) | Tap position of transformer | Capacitive reactive power (Kvar) | Losses (Kw) | Emission (Kg/hr) | Cost of Generators (\$/hr) |
|------------------|---------------------|---------------------------|-----------------------------|----------------------------------|-------------|------------------|----------------------------|
| 15 | 2.95 | 113.50 | 0.98 | | | | |
| 11 | 20.97 | 59.21 | 0.99 | 25.89 | | | |
| 23 | 36.36 | 31.96 | 0.96 | 49.71 | 9.98 | 2.19 | 941.71 |
| | | 27.03 | 0.96 | | | | |
| | | 24.68 | 0.94 | | | | |
| | | 19.63 | | | | | |

CASE 5: The emission produced by FCPPs , the cost of electrical energy generated by FCPPs , active power generation cost and the total electrical energy losses are minimized by varying the generation of generators, the rating and location of FCPPs, capacitive reactive power and tap of transformers, when the load is increased by 120%. The power flow in the line 1-2 is 37.66 MW. Optimizing all the four objectives together, the results obtained are slightly less when compared to optimizing two objectives alone and is given in table 6

Table 6. Results for minimization of emission, cost and power losses

| Location of FCPP | Rating of FCPP (KW) | Rating of Generator (MW) | Tap position of transformer | Capacitive reactive power (Kvar) | Losses (Kw) | Emission (Kg/hr) | Cost of Generators (\$/hr) | Cost of FCPP's (\$/hr) | Total cost (\$/hr) |
|------------------|---------------------|--------------------------|-----------------------------|----------------------------------|-------------|------------------|----------------------------|------------------------|--------------------|
| 15 | 6.24 | 133.50 | 0.92 | | | | | | |
| 10 | 14.32 | 39.21 | 0.94 | 29.22 | | | | | |
| 14 | 33.46 | 31.43 | 0.98 | 41.49 | 10.10 | 2.28 | 953.38 | 2160.8 | 3114.18 |
| | | 29.03 | 0.98 | | | | | | |
| | | 25.78 | 0.96 | | | | | | |
| | | 18.32 | | | | | | | |

6 Conclusion

In this paper, a PSO optimization technique has been used to Optimal Operation Management (OOM) problem in transmission system with Fuel Cell Power Plants (FCPPs).The proposed method was implemented and tested on IEEE 30 bus system and it can be concluded that total emissions, total power losses and total cost of electrical energy are minimized. So this can be extended as a multi-objective optimization problem to obtain a set of Pareto optimal solution allowing the operator choice to use the solution for implementation.

References

1. Zangeneh, A., Jadid, S., Rahimi-Kian, A.: Promotion strategy of clean technologies in distributed generation expansion planning. *Renewable Energy* 34, 65–73 (2009)
2. Dicorato, M., Forte, G., Trovato, M.: Environmental-constrained energy planning using energy-efficiency and distributed-generation facilities. *Renewable Energy* 33, 297–313 (2008)
3. Soroudi, A., Ehsan, M., Zareipour, H.: A practical eco-environmental distribution-network planning model including fuel cells and non-renewable distributed energy resources. *Renewable Energy* 36, 79–88 (2011)
4. Moreira, M.V., da Silva, G.E.: A practical model for evaluating the performance of proton exchange membrane fuel cells. *Renewable Energy* 34, 34–41 (2009)
5. Ackerman, T., Anderson, G., Soder, L.: Distributed generation: a definition. *Electric Power Syst. Res.* 57, 195–204 (2001)
6. Niknam, T., ZeinoddiniMeymand, H., Nayeripour, M.: A practical algorithm for optimal operation management of distribution network including fuel cell power plants. *Renewable Energy* 35, 696–714 (2010)
7. El-Sharkh, M.Y., Tanrioven, M., Rahman, A., Alam, M.S.: Cost related sensitivity analysis for optimal operation of a grid-parallel PEM fuel cell power plant. *J. Power Sources* 161, 198–207 (2006)
8. Mallipeddi, R., Jeyadevi, S., Suganthan, P.N., Bhaskar, S.: Efficient constraint handling for Optimal Reactive Power Dispatch Problem. In: *Swarm and Evolutionary Computation*, vol. 5, pp. 28–36 (2012)
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *IEEE International Conf. on Neural Networks*, Piscataway, NJ, vol. 4, pp. 1942–1948 (1995)

PID Tuning and Control for 2-DOF Helicopter Using Particle Swarm Optimization

A.P.S. Ramalakshmi, P.S. Manoharan, and P. Deepamangai

Department of Electrical and Electronics Engineering, Thiagarajar College of Engineering,
Madurai, India.

ramalakshmi.aps@gmail.com, {psmeee, deepamangai}@tce.edu

Abstract. This paper proposes a proportional-integral-derivative (PID) design of nonlinear multi-input multi-output (MIMO) system called 2 degree-of-freedom (2-DOF) helicopter using particle swarm optimization (PSO) algorithm. The control of 2-DOF helicopter is very challenging task due to its high nonlinearity and complexity. The objective of the control is to control aerodynamic force of 2-DOF helicopter by varying the speed of the pitch and yaw motor and thus tracking their reference position. Statistical measurement and convergence analysis is done for the optimization of gain parameters of the PID controller of 2-DOF helicopter using PSO, modified PSO (MPSO) and genetic algorithm (GA) for equal repetitions of the function evaluation by iteratively minimizing integral of squared error (*ISE*), integral of time multiplied by the squared error (*ITSE*) for 25 independent trials. The numerical simulation results analysis shows the effectiveness of MPSO and PSO compared to GA in controlling the positions of 2-DOF helicopter with consistent tracking.

Keywords: Modified particle swarm optimization (MPSO), 2-DOF helicopter, cross-coupled PID controller.

1 Introduction

The experimental setup of helicopter model having two propellers driven by DC motors mounted on a fixed base is called as 2-DOF helicopter system is considered as a control problem in this correspondence. The elevation of the helicopter nose is controlled by front propeller about pitch axis. The side to side motions can be controlled by back propeller about yaw axis. 2-DOF helicopter is similar to helicopter in certain aspects of behavior. The significant applications of helicopter and its technical importance for the development of unmanned air vehicle leads to the interest of research in the control of helicopter. Many literatures are reported in the trajectory control and it is an active research area [1-3]. Fuzzy supervisory controller is designed for a small quadrotor helicopter in [4]. The control task of helicopter is very tedious and challenging task due to its high-order nonlinearity. However, helicopter possesses great advantage because of its hovering capability and is used

extensively in accessing hard to reach places during search and rescues operations, damage assessment and control in case of natural calamities and accidents involving hazardous environments. Research work is still going on in the quick, versatile control, agility and stability of helicopters [5]. Since high requirement of sophistication is needed in all of the applications, control of helicopter has to be developed efficiently. The control technique is based on the PID control algorithm. Mostly PID controller strategy is used in all industries for process control due to its simplicity [6,7]. PID method of strategic control is the technique currently preferred by most researchers and mentioned in a many literatures [8,9]. Research works are going on to find the key methodologies to tune the gain parameters of the PID controller to have perfect feedback control.

AI is extensively used by many researchers to tune gain parameters of PID controller [10,11]. The PSO is used for the tuning of decentralized typical fuzzy PI controller for the stability of the helicopter model in [12]. The optimal pole locations in state feedback control is searched using PSO to satisfy transient and steady state performance specifications for the rationalized helicopter model in [13]. Recently many research focus is on the design of PID controller by evolutionary algorithms to examine the best optimal value of gain parameters by minimizing the objective function iteratively [14]. These methods are used when aiming for high efficiency in reaching the global optimal solution in the problem search space. GA has more potential for global optimization and is preferred over other methods for most control system problems [15]. Even then the computational time and resources create more overhead than the results obtained. GA was further enhanced by removing the need for binary coding and using real valued parameters in numerical function optimization. New research has exposed the drawbacks in the GA efficiency in spite of the prevalent success that GA has achieved over the past years in solving highly nonlinear optimization problems. GA is especially inefficient when trying to converge closely related parameters. Mutation and crossover operators fail miserably to create a different strain of chromosome from inter related parents.

Kennedy and Eberhart introduced the PSO which is one of the heuristic algorithm [16,17]. The mathematical model is constructed based on the social behavior of birds flocking and fish schooling. Many researchers are working on PSO to enhance its performance to reach global optimum solution by introducing variants and new concepts to improve the searching direction [18]. PSO is extensively used to control highly non-linear systems and achieve stability with very low steady state error. In this paper, statistical measurement and convergence analysis is evaluated for the optimization of gain parameters of the PID controller for the position control of 2-DOF helicopter using PSO, MPSO and GA by iteratively minimizing ISE , $ITSE$.

2 2-DOF Helicopter System

The 2-DOF Helicopter shown in Fig. 1 manufactured by Quanser was used for the experimental test. The helicopter model consists of two propellers which controls the rotational movement about pitch and yaw axis on a fixed base. The two propellers are

driven by DC motors. The elevation of the helicopter nose about the pitch axis is controlled by front propeller and the side to side motions of the helicopter about the yaw axis is controlled by back propeller. Pitch and yaw angles are measured by high-resolution encoders. The pitch encoder and motor signals are transmitted through a slip ring.

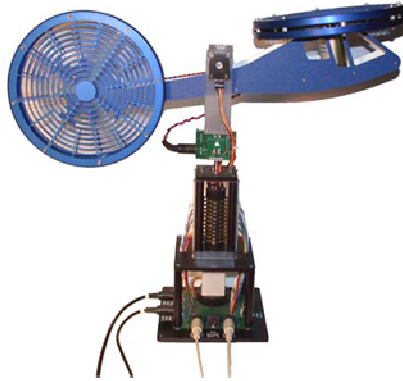


Fig. 1. 2-DOF Helicopter system

2.1 Model

The two degree of freedom helicopter rotates about the pitch axis by angle α_p and about the yaw axis by angle α_y . As shown in Fig. 2, when the nose of the helicopter goes up, the pitch is defined positive and for a clockwise rotation, the yaw is defined positive. The thrust force F_p acting on the pitch axis is normal to the plane of the front propeller and a thrust force F_y acting on the yaw axis is normal to the back propeller. Therefore, at a distance d_p from the pitch axis, a pitch torque is being applied and a yaw torque is applied at a distance d_y from the yaw axis. At the centre of mass of helicopter, the gravitational force F_g generates a torque which pulls down the helicopter nose. The center of mass is a distance of l_{cm} from the pitch axis. The nonlinear equations of motion of 2-DOF helicopter is given as [20],

$$\begin{aligned}
 (J_{eq,p} + m_{heli}l_{cm}^2)\ddot{\alpha}_p &= K_{pp}V_{m,p} + K_{py}V_{m,y} - m_{heli}gl_{cm} \cos\alpha_p - B_p\dot{\alpha}_p \\
 &\quad - m_{heli}l_{cm}^2 \sin\alpha_p \cos\alpha_p \dot{\alpha}_y^2
 \end{aligned}
 \tag{1}$$

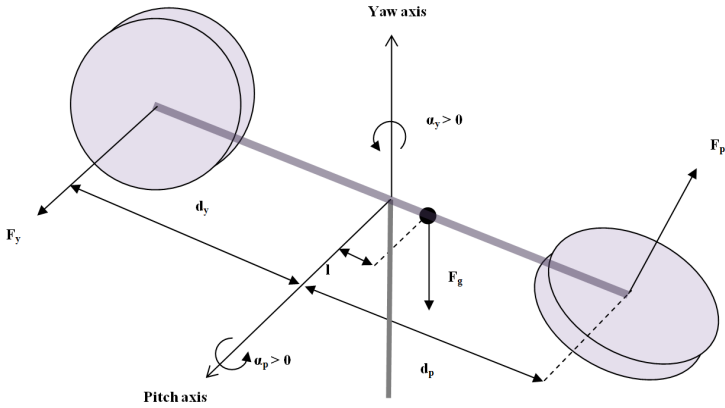


Fig. 2. 2-DOF Helicopter system dynamics

$$\begin{aligned}
 (J_{eq,y} + m_{heli}l_{cm}^2 \cos^2 \alpha_p) \ddot{\alpha}_y &= K_{yy}V_{m,y} + K_{yp}V_{m,p} - B_y \dot{\alpha}_y \\
 &+ 2m_{heli}l_{cm}^2 \sin \alpha_p \cos \alpha_p \dot{\alpha}_y \dot{\alpha}_p
 \end{aligned}
 \tag{2}$$

where $V_{m,p}$ and $V_{m,y}$ are the pitch and yaw motor voltage.

Table 1. Actuator specifications and model parameters

| Symbol | Description | Value | Unit |
|-----------|--|-----------------------|-------------------|
| $J_{m,p}$ | Moment of inertia of pitch motor rotor | 1.91×10^{-6} | kg.m ² |
| $J_{m,y}$ | Moment of inertia of yaw motor rotor | 1.37×10^{-4} | kg.m ² |
| K_{pp} | Thrust torque constant acting on pitch axis from pitch motor/propeller | 0.204 | N.m/V |
| K_{yy} | Thrust torque constant acting on yaw axis from yaw motor/propeller | 0.072 | N.m/V |
| K_{py} | Thrust torque constant acting on pitch axis from yaw motor/propeller | 0.0068 | N.m/V |
| K_{yp} | Thrust torque constant acting on yaw axis from pitch motor/propeller | 0.0219 | N.m/V |

Table 2. 2-DOF Helicopter specifications and model parameters

| Symbol | Description | Value | Unit |
|------------|---|--------|------|
| B_p | Equivalent viscous damping about pitch axis | 0.800 | N/V |
| B_y | Equivalent viscous damping about yaw axis | 0.318 | N/V |
| m_{heli} | Total moving mass of the helicopter | 1.3872 | kg |
| l_{cm} | Center of mass length along helicopter body from pitch axis | 0.186 | m |

Table 2. (continued)

| | | | |
|--------------|--|--------|-------------------|
| $J_{body,p}$ | Moment of inertia of helicopter body about pitch axis | 0.0123 | kg.m ² |
| $J_{body,y}$ | Moment of inertia of helicopter body about yaw axis | 0.0129 | kg.m ² |
| J_{shaft} | Moment of inertia of metal shaft about yaw axis at end point | 0.0039 | kg.m ² |
| J_p | Moment of inertia of front motor/shield assembly about pitch pivot | 0.0178 | kg.m ² |
| J_y | Moment of inertia of back motor/shield assembly about yaw pivot | 0.0084 | kg.m ² |
| $J_{eq,p}$ | Total moment of inertia about pitch pivot | 0.0384 | kg.m ² |
| $J_{eq,y}$ | Total moment of inertia about pitch pivot | 0.0432 | kg.m ² |
| g | Gravitational constant | 9.81 | m/s ² |

The equivalent moment of inertia about the center of mass in Eq. (1) and (2) equals

$$J_{eq,p} = J_{m,p} + J_{body,p} + J_p + J_y \quad (3)$$

$$J_{eq,y} = J_{m,y} + J_{body,y} + J_p + J_y + J_{shaft} \quad (4)$$

where $J_{m,p}$ and $J_{m,y}$ are the moment of inertias of the motor rotor [20].

See Table 1 and Table 2 for the values.

3 PSO Features Overview

Edward and Kennedy designed the framework of PSO in 1995. The basic idea of the algorithm is social behavior of animals, such as bird flocking or fish schooling [17]. PSO starts with an initial set of random population matrix. Each row of population matrix is called particles. The dimension of variable differs according to the problem. Each particle updates its velocity to move to the new position in the cost surface. The particle changes its position based on the velocity and position update equations according to the knowledge of the local and global best solutions as given in Eqs. (5)-(6).

$$v_{id}^{t+1} = v_{id}^t + c_1 * r_1 (pbest_{id}^t - x_{id}^t) + c_2 * r_2 (gbest_{id}^t - x_{id}^t) \quad (5)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, i = 1, 2, \dots, n \quad (6)$$

where c_1 and c_2 are the cognitive and social component. r_1 and r_2 , the random numbers in the range of [0,1] are generated to improve the acceleration of particle's direction to attain the minimum of objective function based on cognitive and social

parameters. v_{id}^t is the particle's velocity along dimension d in t th iteration number which is to be calculated for every generation to move the particle position from existing to new position. $pbest_{id}^t$ is the best solution of that particular particle for a dimension associated with the low cost. $gbest_{id}^t$ is the best solution among all particles in a generation. x_{id}^t is the particle position.

The original PSO is improved by modifying Eq. (5) to

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 * r_1 (pbest_{id}^t - x_{id}^t) + c_2 * r_2 (gbest_{id}^t - x_{id}^t) \tag{7}$$

In Eq. (7), ω is the inertia factor which balances the exploration and exploitation capability of the PSO search algorithm. To accomplish this, value of ω is iteratively reduced from the start to the end of generation.

4 Modified PSO

The standard PSO is modified in such a way that the cognitive component is divided into two factors. The first factor is the previously visited best position of the particle called $pbest_{id}^t$ and the second factor is previously visited worst position of the particle called $pworst_{id}^t$. This modified PSO improves the local exploration of the particle by introducing the $pworst_{id}^t$ to the velocity equation to find its optimal value.

$$v_{id}^{t+1} = \omega v_{id}^t + c_{1b} * r_1 (pbest_{id}^t - x_{id}^t) + c_{1w} * r_2 (x_{id}^t - pworst_{id}^t) + c_2 * r_3 (gbest_{id}^t - x_{id}^t) \tag{8}$$

where, c_{1b} is the acceleration constant which drives the particle toward its best position and c_{1w} is the acceleration constant which drives the particle away from its worst position, r_1 , r_2 and r_3 , the random numbers generated in the range of [0,1]. The particle position is updated using Eq. (6). The exploration capacity of each particle in the swarm is improved locally by Modified PSO.

5 PID Controller Design of 2-DOF Helicopter Using PSO and MPSO

The general PID control law is of the form,

$$u(t) = K_p * e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \tag{9}$$

To control the vertical and horizontal position of highly nonlinear MIMO system like 2-DOF helicopter, cross-coupled PID control scheme is employed as shown in Fig. 3.

It consists of four PID controllers PID_{vv} , PID_{vh} , PID_{hv} and PID_{hh} . These four PID controllers are used to compensate the existence of coupling effect between pitch and yaw actuators [19]. e_v and e_h are the horizontal and vertical position error which is the difference between the setpoint (r_v and r_h) and system response. u_v and u_h are the pitch and yaw motor controlled voltage signal. For the position control of 2-DOF helicopter, the PID controller structure is given as

$$u_v = PID_{vv} + PID_{hv} \quad (10)$$

$$u_h = PID_{hh} + PID_{vh} \quad (11)$$

The design variables for this system are

$$K = [K_{p_{vv}}, K_{i_{vv}}, K_{d_{vv}}, K_{p_{vh}}, K_{i_{vh}}, K_{d_{vh}}, K_{p_{hv}}, K_{i_{hv}}, K_{d_{hv}}, K_{p_{hh}}, K_{i_{hh}}, K_{d_{hh}}] \quad (12)$$

$$= [-20, 20]$$

Two objective functions given in Eqs. (13) to (14) is minimized iteratively to tune the gain parameters K of the cross-coupled PID controller for 2-DOF helicopter using PSO.

$$ISE = \int_0^T [r(t) - y(t)]^2 dt = \int_0^T [e(t)]^2 dt \quad (13)$$

$$ITSE = \int_0^T t \times [r(t) - y(t)]^2 dt = \int_0^T t \times e^2(t) dt \quad (14)$$

where T is total time of integration.

The design of cross-coupled PID controller for 2-DOF helicopter using PSO is summarized as follows:

Data given to the PSO based 2-DOF helicopter system are: Nonlinear 2-DOF helicopter system Eqs. (1) and (2), particle size, dimension, constants c_1 , c_2 are both assigned the value 2, ω of Eq. (7) which is linearly decreased from 0.9 to 0.4 during the course of a run, number of iterations.

The gain parameter of cross-coupled PID controller structure for the 2-DOF helicopter system of Eqs. (1) and (2) is tuned using PSO algorithm such that the objective function of Eqs. (13) and (14) are minimized individually.

- Step 1: Randomly generate the particles K .
- Step 2: Check for maximum generation to stop function evaluation.
- Step 3: Evaluate the objective function Eqs. (13) and (14) for all particles, store previous best position of each particle and global best position among all particles.
- Step 4: Find particle's new velocity by Eq. (7). Update the particle position Eq. (6) by updating the particle velocity equation.

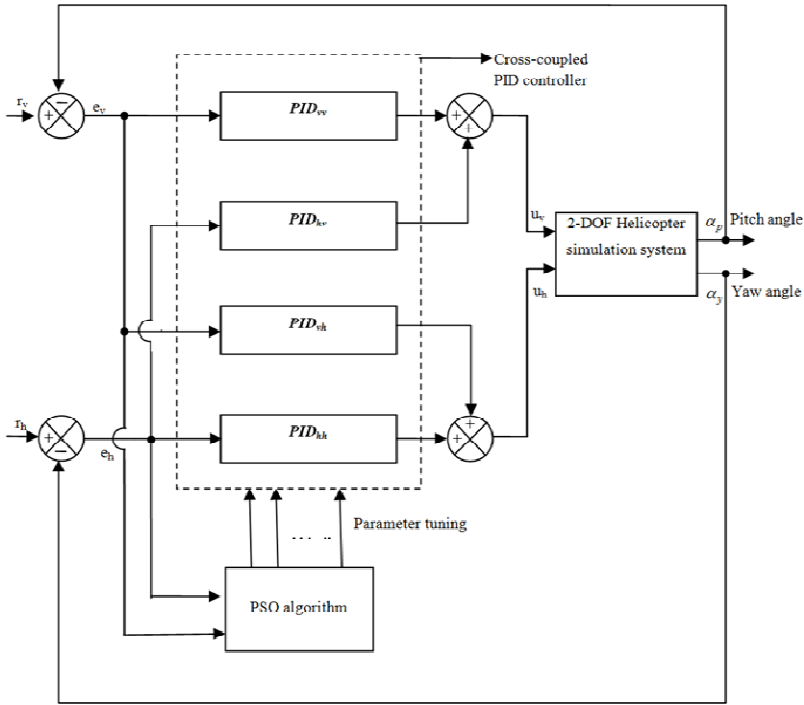


Fig. 3. Block diagram of PSO-PID controller design for the position control of helicopter in 2-DOF

- Step 5: Check whether particle position is within bound. If it is out of bound, do not evaluate such particle.
- Step 6: Go to Step 2.

The same procedure is repeated for MPSO except while calculating the particle's new velocity $pworst_{id}^t$ in Eq. (8) personal worst position is taken into consideration to steer the new position away from the personal worst position.

6 Results and Discussion

We evaluated the objective function Eqs. (13) and (14) iteratively for same number of function evaluations, to analyze the performance of PSO, MPSO, GA algorithms. Best optimal, worst optimal, mean and standard deviation are calculated to analyze the search behavior. In order to implement the GA-PID controller, population matrix K , population size of 50, mutation rate 0.025, selection of 0.5, extrapolation crossover are used.

The block diagram of PSO-PID controller design of helicopter in 2-DOF is shown in Fig. 3. The PSO-PID controller is tuned to obtain best optimal gain parameter K

under two performance criteria. The convergence analysis of PSO,MPSO and GA algorithm for helicopter in 2-DOF by minimizing ISE , $ITSE$ for 500 generations, 25 trials is shown in Fig. 4. The sine input, of 11.46 degree and, 0.05 Hz and the sine input of 28.65 degree and, 0.05Hz is given as the reference signal to test the output position response α_p and α_y of helicopter. The simulation is run from 0 to 50 seconds with the sampling time of 0.02 seconds. The Fig. 5 shows the simulated pitch sine response, yaw sine response for the best optimal result which is obtained out of 25 trials by minimizing $ITSE$. The convergence and statistical analysis of all three algorithms are summarized in Table 3. The simulated pitch and yaw response shows consistent tracking in set point with less steady state error. The experimental observation is analyzed by applying the gain parameters to the gain block of cross-coupled PID controller. The obtained simulation results are compared with the experimental results to analyze the performance of 2-DOF helicopter. The pitch angle and yaw angle are observed in this closed-loop implementation. The actual response shows tracking of simulated response with deviations. This is due to the existence of coupling effect between pitch and yaw actuators and uncertainty in the actuators and model parameter specifications of 2-DOF helicopter.

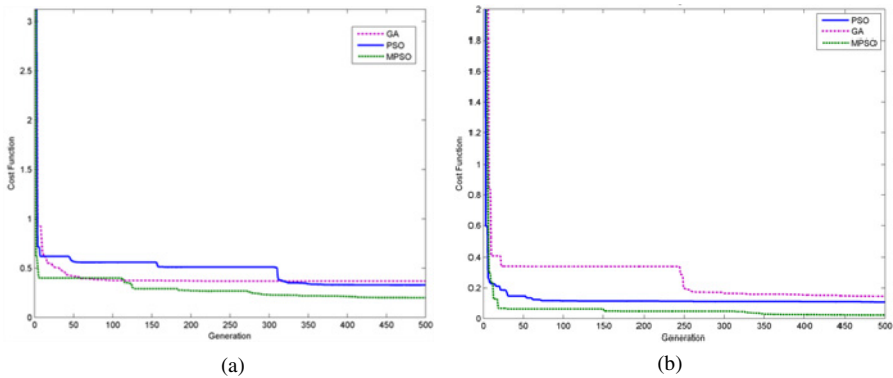


Fig. 4. Convergence analysis for different cost function (a) ISE (b) ITSE

Table 3. Statistical analysis

| Objective function | Algorithms | Number of generations | Results | | | |
|--------------------|------------|-----------------------|------------|-------------|------------|--------------------|
| | | | Best value | Worst value | Mean value | Standard deviation |
| ITSE | MPSO | 500 | 0.0222 | 0.0623 | 0.0364 | 0.0114 |
| | PSO | 500 | 0.1076 | 0.5184 | 0.1705 | 0.0841 |
| | GA | 500 | 0.1457 | 0.9267 | 0.3695 | 0.3090 |
| ISE | MPSO | 500 | 0.1995 | 0.2304 | 0.2116 | 0.0089 |
| | PSO | 500 | 0.3281 | 0.8634 | 0.4074 | 0.1108 |
| | GA | 500 | 0.3665 | 1.0611 | 0.5881 | 0.2068 |

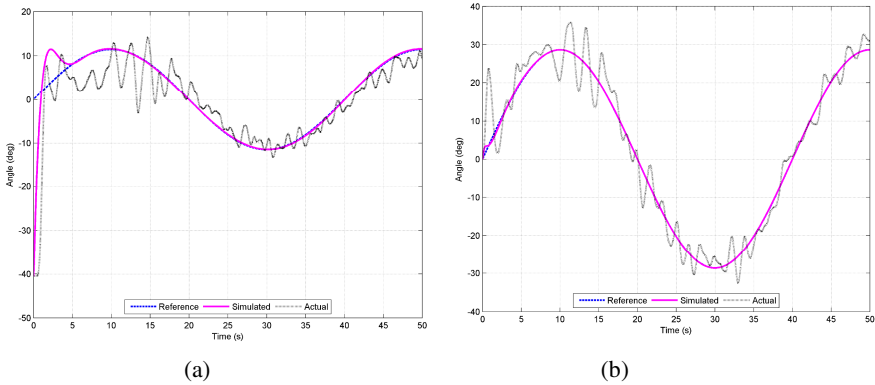


Fig. 5. Sine response for pitch motor and yaw motor

7 Conclusion

In this paper, cross-coupled PID controller is designed by minimization of *ISE* , *ITSE* using PSO for a highly nonlinear dynamic system like 2-DOF helicopter. The performance of MPSO, PSO, GA is compared in terms of statistical measure and convergence behavior over same number of function evaluations. The convergence analysis of the two algorithms show the effectiveness of MPSO in searching best optimal value to tune gain parameters *K* and to track the desired reference signal. The search behavior of PSO is examined by performing statistical analysis and the numerical gain parameters of the cross-coupled PID controller are calculated corresponding to best optimal value for two performance criteria. The simulation results show the effectiveness of MPSO in tuning the nonlinear system like 2-DOF helicopter with less steady state error and good tracking of command signal. The position control of 2-DOF helicopter is experimentally done and observed using cross-coupled PID controller.

PSO is a powerful technique in tuning the parameters for the PID controller. However, in a nonlinear system like 2-DOF helicopter maintaining control is very tedious. So in future to improve control, design of robust PID controller using PSO can be considered to improve the performance of the nonlinear system in the presence of uncertainty and external disturbances.

Acknowledgments. One of the authors of this paper (P.S. Manoharan) acknowledges Science and Engineering Research Board, Department of Science and Technology (DST), India for sanctioning the funding under Fast Track Young Scientist Scheme, vide sanction number SERB/F/2056/11-12 dated 15.02.2012.

References

1. Ding, X., Yu, Y.: Motion Planning and Stabilization Control of a Multipropeller Multifunction Aerial Robot. *IEEE/ASME Transactions on Mechatronics* 18(2), 645–656 (2013)

2. Mahmoud, M.S., Koesdwiady, A.B.: Improved digital tracking controller design for pilot-scale unmanned helicopter. *Journal of the Franklin Institute* 349, 42–58 (2012)
3. Chudy, P., Dittrich, P., Rzucidlo, P.: HIL simulation of a light aircraft flight control system. In: *IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, Williamsburg, VA, pp. 14–18 (2012)
4. Hosseininasab, H., Jafarboland, M.: Design a Fuzzy Supervisory Controller for a Small Quadrotor Helicopter. *International Review of Automatic Control* 4(3), 472–480 (2011)
5. Wen, P., Wen, P., Lun, T.W.: Decoupling control of a twin rotor MIMO system using robust deadbeat control technique. *IET Control Theory Appl.* 2(11), 999–1007 (2008)
6. Juang, J.-G., Huan, M.-T., Liu, W.-K.: PID Control Using Presearched Genetic Algorithms for a MIMO System. *IEEE Trans. Syst. Man Cybernet. Part C: Applications and Reviews* 38(5), 716–727 (2008)
7. Astrom, K.J., Hagglund, T.: *PID controllers: Theory, Design and Tuning*. Instrument Society of America (1995)
8. Ziegler, J.B., Nichols, N.B.: Optimum settings for automatic controllers. *ASME Trans.* 64, 759–768 (1942)
9. Kao, C.C., Chuang, C.W., Fung, R.F.: The self-tuning PID control in a slider-crank mechanism system by applying particle swarm optimization approach. *Mechatronics* 16(8), 513–522 (2006)
10. Subbu, R., Goebel, K., Frederick, D.K.: Evolutionary design and optimization of aircraft engine controllers. *IEEE Trans. Syst. Man Cybernet. Part C: Applications and Reviews* 35(4), 554–565 (2005)
11. Belarbi, K., Titel, F.: Genetic algorithm for the design of a class of fuzzy controllers: An alternative. *IEEE Trans. Fuzzy Syst.* 8(3), 398–405 (2000)
12. Boubertakh, H., Labiod, S., Tadjine, M.: PSO to design decentralized fuzzy PI controllers application for a helicopter. In: *Mediterranean Conference on Control & Automation (MED)*, pp. 633–637 (2012)
13. Wang, J., Brackett, B.T., Harley, R.G.: Particle Swarm-assisted state feedback control: From pole selection to state estimation. In: *American Control Conference*, pp. 1493–1498 (2009)
14. Haupt, R.L., Haupt, S.E.: *Practical genetic algorithms*, 2nd edn. John Wiley & Sons, Inc., New Jersey (2004)
15. Michalewicz, Z., Dasgupta, D.: *Evolutionary Algorithms in Engineering Applications*. Springer (1997)
16. Eberhart, R.C., Kennedy, J.: A new optimizer using particles swarm theory. In: *Proceedings of 6th International Symposium on Micro Machine and Human Science*, pp. 39–43 (1995)
17. Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 69–73 (1998)
18. Deepa, S.N., Sugumaran, G.: Model order formulation of a multivariable discrete system using a modified particle swarm optimization approach. *Swarm and Evolutionary Computation* 1(4), 201–212 (2011)
19. Ramalakshmi, A.P.S., Manoharan, P.S.: Non-linear modeling and PID control of twin rotor MIMO system. In: *Proceedings of IEEE Conference on Advanced Communication Control and Computing Technologies*, pp. 366–369 (2012)
20. Quanser innovate educate, Aerospace plant: 2-DOF Helicopter position control manual, Canada (2006)

Optimal Location and Parameter Selection of Thyristor Controlled Series Capacitor Using Particle Swarm Optimization

S. Devi^{1,*} and M. Geethanjali²

¹Department of EEE,
National College of Engineering, Tirunelveli-627 151, India
devi_balasubramanian@yahoo.com

²Department of EEE,
Thiagarajar College of Engineering, Madurai-625 015, India

Abstract . This paper proposes the optimal location of Thyristor Controlled Series Capacitor (TCSC) to reduce the active power loss in radial distribution system using Particle Swarm Optimization (PSO) algorithm. Also in this work, reactance characteristics curve of a TCSC device is drawn between effective reactance of TCSC (X_{TCSC}) and firing angle ' α ' from the optimal output obtained from the PSO algorithm. This gives the effective value of the firing angle ' α ' and the main parameters of TCSC (such as X_C and X_L) are also designed. This analysis is done by degree of series compensation ' k '. The feasibility and effectiveness of the optimization methods proposed have been demonstrated on 12, 34 and 69 bus radial distribution system consisting of 11, 33 and 68 sections respectively. MATLAB, Version 7.10 software is used for simulation.

Keywords: Particle Swarm Optimization (PSO), Direct Load Flow (DLF) Method, Radial Distribution System (RDS), Thyristor Controlled Series Capacitor (TCSC), Degree of Series Compensation ' k ', Flexible AC Transmission systems (FACTS).

1 Introduction

From the generating station the power is distributed to the consumers by the use of transmission and sub transmission units through the distribution lines. The voltage profile improvement and system losses reduction depends on the placement of capacitors [1]. The power transfer parameters like transmission voltage, line impedance and phase angle, are dynamically controlled with FACTS devices that play the major role in power electronics based technology [2]. FACTS devices facilitate an increase in flexibility, lower operation and maintenance costs with less environment externalities [3]. The voltage profile improvement and the energy losses are reduced in the radial distribution system by the installation of capacitors at suitable locations. The active power (I^2R) losses are reduced by 13 % in the distribution networks from the total power generation [4].

* Corresponding author.

To solve optimization problems, Evolutionary Algorithms proposed Differential Evolution (DE) [5]. As traditional EAs, DE solved many optimization problems successfully [6]. Genetic Algorithm (GA) is also used to optimize, plan reactive power, evaluate system losses and also to find the optimal location of FACTS devices [7]. For FACTS based controller design, the application and performance comparison of PSO and GA optimization techniques are discussed [8]. Particle Swarm Optimization (PSO) which is one more evolutionary computation method could be used for solving the problem of FACTS devices sizing and allocation. In recent days, the application of PSO is suitable for the optimal placement of FACTS devices in power system [9].

From this literature survey it is known that, the optimum placement of FACTS device is efficiently done by the Evolutionary Algorithms. For this work PSO algorithm is considered to find the optimal allocation and sizing of a Thyristor Controlled Series Capacitor (TCSC) for series compensation in a power system. The criterion should be set in such a way to find the to optimize the voltage profile of the system and the TCSC size such that the real power, voltage deviations at each bus, current and effective reactance of the line do not exceed a predefined set value.

Also in this work, from the optimal X_{TCSC} value obtained from the PSO algorithm, the reactance characteristics curve of a TCSC device is drawn between effective reactance of TCSC and firing angle ' α '. This gives the effective value of the firing angle ' α ' and the main parameters of TCSC (such as X_C and X_L) are also designed. Therefore this work consists of two parts.

Part-1 (Using PSO)

- Optimum location of TCSC
- Sizing of TCSC (X_{TCSC})

Part-2 (Using Degree of Series Compensation 'k')

- Selection of firing angle ' α '
- Design of parameters X_C and X_L

2 Flexible AC Transmission System (FACTS)

In 1986 N. G. Hingorani invented FACTS Technology for the amendments of usual electrical characteristics of ac power system. FACTS controllers are classified as series and shunt controllers. Using series compensation, transmission or distribution system parameters are customized. Series controllers increase the power handling capacity. The power transmission capability of the lines can be improved by the series compensation which is an economic method [10, 11]. By shunt compensation technique the equivalent impedance of the load can be changed. Shunt controllers improve the voltage at a particular location.

2.1 Thyristor Controlled Series Compensator (TCSC)

Many benefits for a power system including controlling power flow in the line, damping power oscillations, and mitigating sub synchronous resonance can be provided by the Thyristor-Controlled Series Capacitors (TCSC). It is also a type of series compensator. TCSC is able to modify the line impedance and thereby control the power flow. The capacitor is inserted directly in series with the transmission line and the thyristor-controlled inductor is mounted in parallel with the capacitor.

Thus no interfacing equipment like high voltage transformers are required. This makes TCSC much more economical simple with ease of operation than some competing FACTS technologies. World’s first 3 phase [12], 2 X 165 MVAR, TCSC was installed in 1992 in Kayenta substation, Arizona. TCSC consists of series compensating capacitor shunted by thyristor controlled reactor. It is modeled as a controllable reactance, inserted in series with the transmission line to adjust the line impedance and thereby control the power flow [13].

2.2 Basic Concepts

The basic operation of TCSC device can be easily explained from a schematic diagram shown in Fig. 1. It consists of a series compensating capacitor shunted by a Thyristor controlled reactor (TCR). TCR is a variable inductive reactor X_L controlled by firing angle ‘ α ’. Fig. 2 shows the reactance characteristics curve of a TCSC device drawn between effective reactance of TCSC and firing angle ‘ α ’. The effective reactance ‘ $X_{TCSC}(\alpha)$ ’ of TCSC operates in three regions: inductive region, capacitive region and resonance region. Inductive region starts increasing from inductive reactance $X_L \parallel X_C$ value to infinity (parallel resonance condition, ‘ $X_L(\alpha) = X_C$ ’), and decreasing from infinity to capacitive reactance X_C for capacitive region. Resonance occurs between the two regions. The reactance characteristics of TCSC operation in both capacitive and inductive regions [14] through variation of firing angle ‘ α ’ is as shown below:

- a) Region 90° to α_{Lim} - Inductive region
- b) Region α_{Clim} to 180° - Capacitive region
- c) α_{Lim} to α_{Clim} - Resonance region

The resonance region is avoided by installing limits in the firing angle. At the resonant point, the TCSC exhibits very large impedance and results in a significant voltage drop. It is clear that the vernier operation of TCSC can only enhance the apparent reactance in both capacitive and inductive domain. Also firing angles close to the resonant point would lead to high reactance operation which is not practical. To avoid high reactance region, maximum and minimum control angles in the inductive and capacitive regions should be established.

2.3 Mathematical Modeling of TCSC

A TCSC is a series type FACTS device inserted for line reactance compensation in the line between ‘i’ and ‘j’ as shown in Fig. 3. TCSC can operate either in inductive mode or in capacitive mode. (+) Sign is for inductive reactance and (-) sign is for capacitive reactance. So the net reactance of the transmission line becomes [11, 12],

$$X_{TOTAL} = X_{ij} \pm X_{TCSC}(\alpha) \tag{1}$$

where ‘ α ’, the firing angle of TCSC varies from 90° to 180° . Effective TCSC reactance (X_{TCSC}) with respect to firing angle (α) can be given as:

$$X_{TCSC}(\alpha) = -X_C + C_1(2(\pi - \alpha) + \sin(2(\pi - \alpha))) - C_2 \cos^2(\pi - \alpha)(\omega \tan(\omega(\pi - \alpha)) - \tan(\pi - \alpha)) \tag{2}$$

Where,

$$C_1 = \frac{X_C + X_L}{\pi} \tag{3}$$

$$C_2 = 4 \frac{1}{X_L \pi} \left(\frac{X_C X_L}{X_C - X_L} \right) \tag{4}$$

$$\omega = \sqrt{\frac{X_C}{X_L}} \tag{5}$$

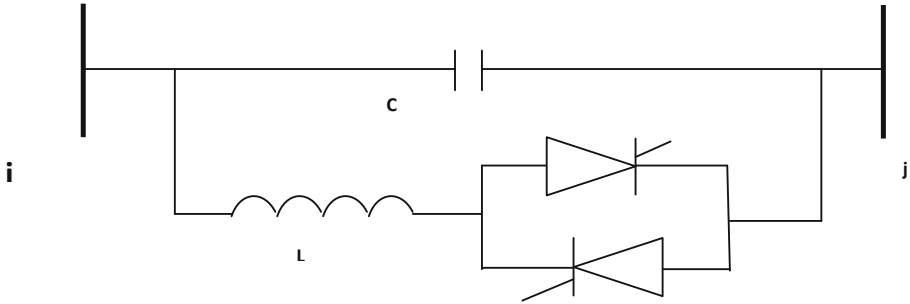


Fig. 1. Schematic diagram of TCSC device

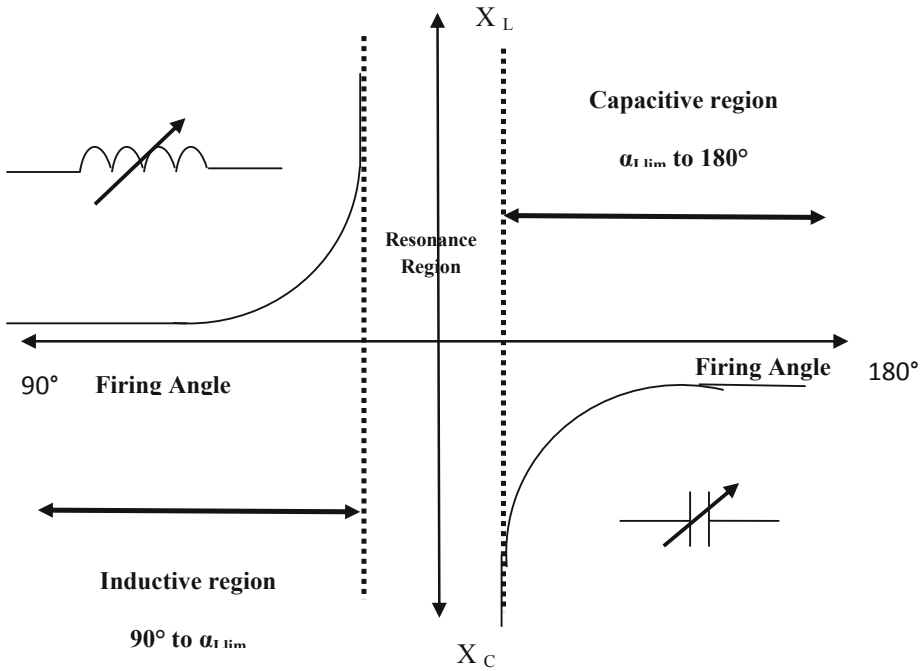


Fig. 2. TCSC's Reactance characteristic curve

$$k = \frac{X_C}{X_L} \quad (6)$$

$$X_{TCSC} = \frac{X_C X_L}{X_C - X_L} \quad (7)$$

where 'k' is the degree of series compensation and X_{ij} is the line reactance between bus-i and bus-j.

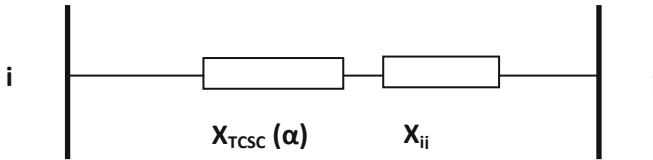


Fig. 3. Line with TCSC

3 Problem Formulation

The objective of the optimal placement and sizing of TCSC is to minimize the active power loss in the distribution network.

The objective function:

$$\min f = \min(P_{T, LOSS}) \quad (8)$$

where,

T_{Loss} is the total power loss of the radial distribution system.

Constraints:

Inequality constraints:

Real power generation constraint

$$P_{gi}^{\min} \leq P_{gi} \leq P_{gi}^{\max} \quad (9)$$

Reactive power generation constraint

$$Q_{gi}^{\min} \leq Q_{gi} \leq Q_{gi}^{\max} \quad (10)$$

Voltage constraints:

$$V_i^{\min} \leq V_i \leq V_i^{\max} \quad (11)$$

Current constraints:

$$I_i^{\min} \leq I_i \leq I_i^{\max} \quad (12)$$

Reactance limit of TCSC:

$$-0.9 * X_{ij} \leq X_{TCSC} \leq 0.9 * X_{ij} \quad (13)$$

where P_g^{\min} and P_g^{\max} —minimum and maximum limits of real power generation, Q_g^{\min} and Q_g^{\max} —minimum and maximum limits of reactive power generation, $|V_i^{\min}|$ and $|V_i^{\max}|$ —minimum and maximum limits of magnitude of bus voltage, $|V|$ —magnitude of bus voltage.

4 Development of Test System

4.1 Load Flow Analysis

Distribution load flow is important for distribution automation systems, and distribution management systems. Network optimization, VAR planning, switching, state estimation etc. need the support of vigorous and efficient load flow solution. Conventional load flow studies like Gauss-seidal and fast decoupled load flow and Newton-Raphson methods are not suitable for distribution system load flows because of high R/X ratio. A Direct Approach for Distribution System Load Flow which offers better solution was proposed in [15]. Using two developed matrices, the bus-injection to branch-current matrix and the bus-current to bus-voltage matrix load flow solutions were obtained with simple matrix multiplication.

4.2 Bus Building Algorithm

For distribution networks, the complex load S_i is expressed as,

$$S_i = P_i + Q_i, \quad i=1,2,\dots,N \tag{14}$$

where N is the total no of buses, P_i is the real power at i th bus and Q_i is the reactive power at the i th bus. Current injection is given as,

$$I_i = (S_i / V_i)^* \tag{15}$$

The relationship between bus current injections and bus voltages are expressed as

$$[\Delta V] = [BCBV][BIBC][I] \tag{16}$$

$$= [DLF][I] \tag{17}$$

where,

BCBV is the branch-current to bus-voltage matrix.

BIBC is the bus-injections to branch-currents matrix.

The solution for radial distribution load flow can be obtained by solving the equations (28-30) iteratively.

$$I_i^k = (S_i / V_i^k)^* \tag{18}$$

$$[\Delta V^{k+1}] = [DLF] [I^k] \tag{19}$$

$$[V^{k+1}] = [Vo] [\Delta V^{k+1}] \tag{20}$$

5 Power Flow Calculation

The power flows are computed by the following set of simplified recursive equations (16-22).

$$P_{i+1} = P_i - P_{Li+1} - R_{Li+1} * \frac{(P_i^2 + Q_i^2)}{|V_i|^2} \tag{21}$$

$$Q_{i+1} = Q_i - Q_{Li+1} - X_{Li+1} * \frac{(P_i^2 + Q_i^2)}{|V_i|^2} \tag{22}$$

Where P_i and Q_i are the real and reactive powers flowing out of bus i , and P_{Li} and Q_{Li} are the real and reactive load powers at bus i . 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 $i + 1$ may be computed as:

$$P_{Loss} (i, i+1) = R_{Li+1} * \frac{(P_i^2 + Q_i^2)}{|V_i|^2} \tag{23}$$

$$Q_{Loss} (i, i+1) = X_{Li+1} * \frac{(P_i^2 + Q_i^2)}{|V_i|^2} \tag{24}$$

The total active power loss of the feeder, $P_{T,LOSS}$, may then be determined by summing up the losses of all line sections of the feeder, which is given as:

$$P_{T,LOSS} = [BIBC] * P_{LOSS} (i, i+1) \tag{25}$$

6 Particle Swarm Optimization (PSO) Algorithm

Particle swarm optimization (PSO) is an evolutionary computation optimization technique (a search method based on a natural system) developed by Kennedy and Eberhart [16]. It is a multiagent search technique and it uses a number of particles that constitute a swarm. Each particle traverses the search space looking for the global minimum (or maximum). The particles fly around in a multidimensional search space and each particle adjusts its position according to its own experience and the experience of neighboring particles making use of the best position encountered by itself and its neighbours. Here the concept of velocity is used to represent the modifications in each particle. Velocity of each particle can be modified by using equation (24).

$$V_i^{k+1} = \omega V_i^k + C_1 rand 1 * (Pbest - S_i^k) + C_2 rand 2 * (Gbest - S_i^k) \tag{26}$$

where,

$$c = \frac{2}{(2 - \phi - \phi1)} \tag{27}$$

$$\phi1 = \sqrt{\phi^2 - 4\phi} \tag{28}$$

$$\omega = \omega_{max} - \frac{[(\omega_{max} - \omega_{min}) * currentgen \text{ erationnum ber}]}{Maximumgen \text{ erationnum ber}} \tag{29}$$

ω_{max} Initial value of the inertia weight

ω_{min} Final value of the inertia weight

Using equation (24), a certain velocity, which gradually gets close to $Pbest$ and $Gbest$ can be calculated. The current position can be modified by using equation (28).

$$S_i^{k+1} = S_i^k + V_i^{k+1} \tag{30}$$

where S^k is current searching point, S^{k+1} is modified searching point, v^k is current velocity, v^{k+1} is modified velocity of agent i , ω is weight function for velocity of the agent, C_1 and C_2 are weight coefficients for each term and rand1 , rand2 are the random value generated between $[0,1]$. Φ is a random number. In this paper, the evolutionary optimization is used to address the X_{TCSC} placement and sizing problem. The particle in the PSO algorithm, as applicable to this problem, consists of randomly initialized X_{TCSC} sizing.

6.1 Implementation of PSO Algorithm

The PSO-based approach for solving the optimal placement and sizing of TCSC problem to minimize the real power loss takes the following steps:

Step 1: Get the Input. The input data are line (Line Impedance) and bus data (Load Power i.e., Real Power and Reactive Power) and bus voltage limits.

Step 2: Calculate the real power loss using distribution load flow based on Direct Load Flow Method.

Step 3: Set the generation counter $j = 0$.

Step 4: With random positions and velocities, randomly generates an initial population.

Step 5: Set the bus count $C=2$.

Step 6: For each particle calculate the real power loss using equation (13).

Step 7: Check the bus voltage lies in the limits or not. If it is not lies in the limit that particle is infeasible.

Step 8: For each particle, compare its objective value with the individual best. If the objective value is lower than P_{best} , set this value as the current P_{best} , and record the corresponding particle position.

Step 9: Choose the particle associated with the minimum individual best P_{best} of all particles, and set the value of this P_{best} as the current overall best G_{best} .

Step 10: Update the velocity and position of particle using equation (16) and (17) respectively.

Step 11: If the bus count reaches the maximum limit, go to step 12. Otherwise, set bus count $C=C+1$, and go back to step 6.

Step 12: If the generation number reaches the maximum limit, go to Step 13. Otherwise, set generation index $j = j + 1$, and go back to Step 4.

Step 13: Print out the optimal solutions.

The solutions include the optimal location and size of TCSC in radial distribution systems. The corresponding fitness values to these solutions indicate the minimum total real power loss.

7 Calculation of Firing Angle (α) and Parameter Selection of TCSC

After finding the optimum place and sizing of TCSC using PSO with real power loss as the objective function, the value of firing angle is calculated as follows.

- i) Using the equation (2-5), reactance characteristics curve of a TCSC device is drawn between effective reactance of TCSC and firing angle α for various k .
- ii) From the best value of k the firing angle for the required X_{TCSC} is obtained.

- iii) Using X_{TCSC} obtained from PSO algorithm, for various values of degree of series compensation 'k', X_L and X_C are calculated using equation (6, 7).

8 Implementation of the Proposed Method

The complete structure of this work to solve the optimal TCSC placement and sizing of the test systems using PSO algorithm is shown in fig.4. At first the power loss is calculated from direct load flow method. After placing the TCSC, the active power loss is calculated using PSO algorithm. The real power loss is minimized, by placing the TCSC in optimum place. The procedure is repeated until no further minimum losses from the TCSC placement are achieved. From X_{TCSC} obtained from PSO algorithm, for various values of 'k', reactance characteristics curve of a TCSC device is drawn between effective reactance of TCSC (X_{TCSC}) and firing angle ' α '. From the best value of ' α ' the main parameters of TCSC such as 'k', X_L and X_C are designed.

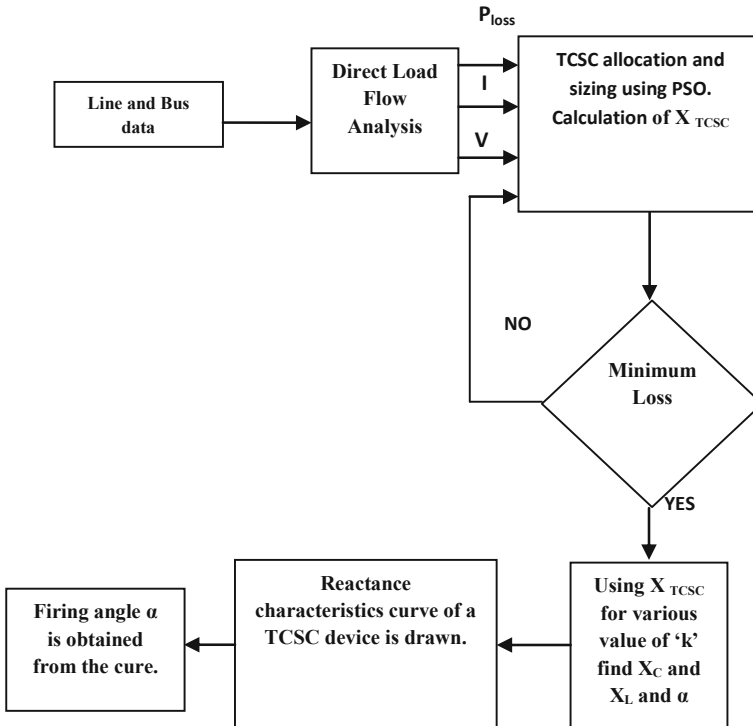


Fig. 4. Flow chart of the entire work

9 Case Studies

9.1 Simulation Results and Analysis

In order to evaluate the proposed work, 12 bus, 34 bus and 69 bus test systems [17] are considered. The total power loss for the 12 bus, 34 bus and 69 bus test systems

obtained from the DLF are 0.0145 MW, 0.1638 MW and 0.24479 MW respectively. The simulated results are tabulated and analyzed using MATLAB 2010.

9.2 PSO Result Analysis

The parameters used to evaluate PSO algorithm are listed in table 1. From table 2, after placement of TCSC, the real power loss is 0.0138, 0.0411 and 0.1990 MW for 12 bus, 34 bus and 69 bus respectively. The optimum branch number for 12 bus, 34 bus and 69 bus is 7, 21 and 56 respectively. It is shown in fig.5 to fig.7. The voltage profile is also improved. The size of the TCSC and the line reactance for the test systems are given in table 2.

Table 1. Selection of parameters for various algorithms used

| Test system | Parameters | | | | | | |
|-------------|------------|----------|------------|----------------|----------------|--------------|--------------|
| | Population | Iteation | C_1, C_2 | ω_{max} | ω_{min} | Φ_{max} | Φ_{min} |
| 12 bus | 25 | 100 | 1.8 | 0.8 | 0.1 | 0.42 | 0.41 |
| 34 bus | 30 | 80 | 1.8 | 0.8 | 0.1 | 0.42 | 0.41 |
| 69 bus | 30 | 80 | 1.8 | 0.8 | 0.1 | 0.42 | 0.41 |

Table 2. Results from PSO algorithm

| | Without TCSC | | | With TCSC | | |
|-----------------------------------|--------------|--------|--------|-----------|---------|--------|
| | 12 bus | 34 bus | 69 bus | 12 bus | 34 bus | 69 bus |
| Active Power Loss (MW) | 0.0145 | 0.1638 | 0.2254 | 0.0138 | 0.0411 | 0.1990 |
| Voltage (p.u) | 0.9613 | 0.9663 | 0.9625 | 0.9791 | 1.008 | 1.0062 |
| Branch | - | - | - | 6-7 | 20-21 | 56-57 |
| Size of TCSC (X_{TCSC}) (ohm) | - | - | - | 26.7433 | 44.0259 | 9.8377 |
| Branch Reactance (ohm) | 0.4170 | 0.043 | 0.5337 | 26.3211 | 44.0695 | 9.3040 |

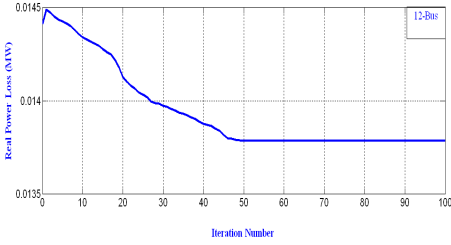


Fig. 5. Iteration number Vs real power loss at 20th branch for 12 bus system

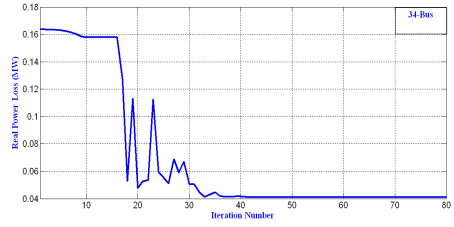


Fig. 6. Iteration number Vs real power loss at 20th branch for 34 bus system

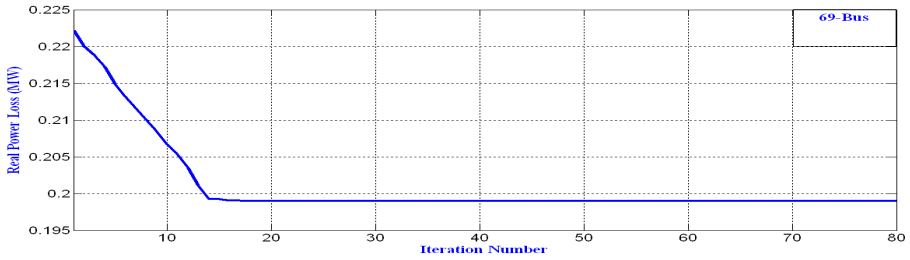


Fig. 7. Iteration number Vs real power loss at 56th branch for 69 bus system

Table 3. Firing Angle and Resonance region for various k

| Test System | k | Starting of Resonance Region (deg) | Ending of Resonance Region (deg) | Resonance Region (deg) | Firing Angle (deg) | Distance between resonance region and firing angle (deg) |
|-------------|------------|------------------------------------|----------------------------------|------------------------|--------------------|--|
| 12-Bus | k=2 | 112.9 | 121.5 | 8.6 | 103.7 | 9.2 |
| | k=3 | 124.4 | 133.2 | 8.8 | 111.3 | 13.1 |
| | k=4 | 130.5 | 138.7 | 8.2 | 117.7 | 12.8 |
| | k=5 | 136 | 144.7 | 8.7 | 123.7 | 12.3 |
| | k=6 | 139 | 147.3 | 8.3 | 128.8 | 10.2 |
| 34-Bus | k=2 | 113.1 | 122.5 | 9.4 | 103.4 | 9.7 |
| | k=3 | 124.4 | 133.3 | 8.9 | 109.8 | 14.6 |
| | k=4 | 129.9 | 138.8 | 8.9 | 116.8 | 13.1 |
| | k=5 | 135.9 | 144.4 | 8.5 | 123.1 | 12.8 |
| | k=6 | 138.8 | 147.3 | 8.5 | 128 | 10.8 |
| 69-Bus | k=2 | 113 | 122.6 | 9.6 | 103.1 | 9.9 |
| | k=3 | 124.4 | 133.3 | 8.9 | 109.8 | 14.6 |
| | k=4 | 129.9 | 138.8 | 8.9 | 116.8 | 13.1 |
| | k=5 | 135.9 | 144.4 | 8.5 | 123.1 | 12.8 |
| | k=6 | 138.8 | 147.3 | 8.5 | 128 | 10.8 |

9.3 Reactance Curve Analysis

The reactance curve between the firing angle ‘ α ’ and the X_{TCSC} for various values of ‘ k ’ is shown in fig.8 to fig.10. Corresponding to the optimal X_{TCSC} obtained from the PSO, the firing angle and the resonance region for different test systems with different ‘ k ’ values are tabulated in table 3. Table 4 shows the maximum variation in capacitive and inductive reactance region for various values of ‘ k ’.

From table 3 and 4, the best value of k is chosen to be 3 because it has the maximum distance between the resonance region and the firing angle which means the firing angle is far away from the resonance region. Always it is not preferable to operate the firing angle nearer to the resonance region.

Table 4. Maximum variation of X_L and X_C

| k | Maximum variation in Capacitive Reactance Region (ohm) | | | Maximum variation in Inductive Reactance Region (ohm) | | |
|---|--|--------|--------|---|--------|--------|
| | 12 Bus | 34 Bus | 69 Bus | 12 Bus | 34 Bus | 69 Bus |
| 2 | 266 | 418.8 | 98.1 | 1069 | 1769 | 399.6 |
| 3 | 420 | 682.9 | 150.7 | 1088 | 1800 | 406.7 |
| 4 | 1144 | 1873 | 423.3 | 457.1 | 758.8 | 171.5 |
| 5 | 620.5 | 1004 | 228.1 | 1015 | 1680 | 379.6 |
| 6 | 991.2 | 1618 | 366.7 | 623.3 | 1032 | 233.3 |
| 7 | 1209 | 1530 | 348.5 | 983.5 | 1160 | 144.4 |

Table 5. Parameters of TCSC with $k=3$

| Test System | X_L (ohm) | X_C (ohm) | Firing Angle (deg) |
|-------------|-------------|-------------|--------------------|
| 12-Bus | 17.82887 | 53.4866 | 111.3 |
| 34-Bus | 29.3506 | 88.0518 | 109.8 |
| 69-Bus | 6.55847 | 19.6754 | 109.8 |

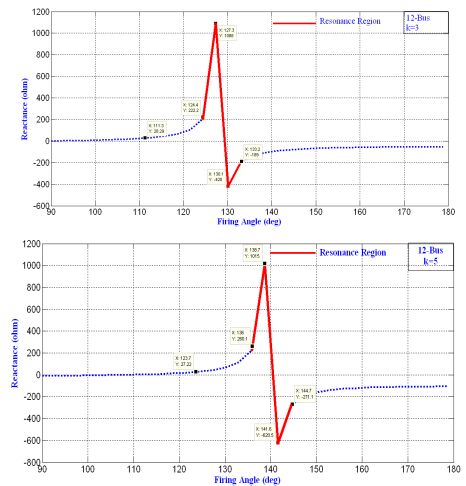
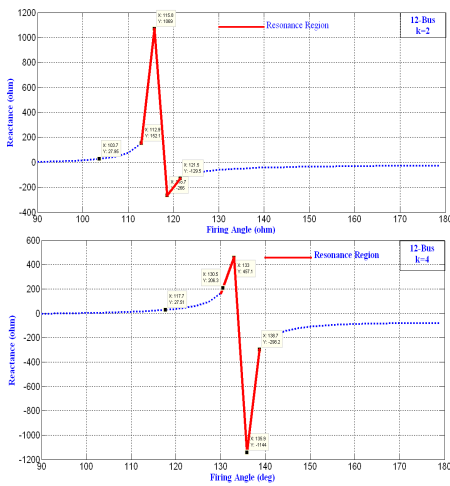


Fig. 8. Reactance curve between firing angle and X_{TCSC} for 12 bus system with various values of ‘ k ’

From table 3 and 4, it is inferred that the maximum value of capacitive reactance corresponding to 'k' equal to 3 is high compared to 'k' equal to 2. Even then, k=3 is chosen to be the best because it has the maximum distance between the firing angle and resonance region which is mostly preferred. The maximum value of capacitive reactance in k=3 can be overlooked when compared to other values of capacitive reactance obtained from the remaining values of 'k'.

Also from the table 3, it is observed that for each value of 'k' the tabulated values show the similarities within the test system. It shows the superlative of this work. From table 4, it is inferred that the maximum variation of capacitance increases with k. The X_L and X_C values with k equal 3 are listed in table 5.

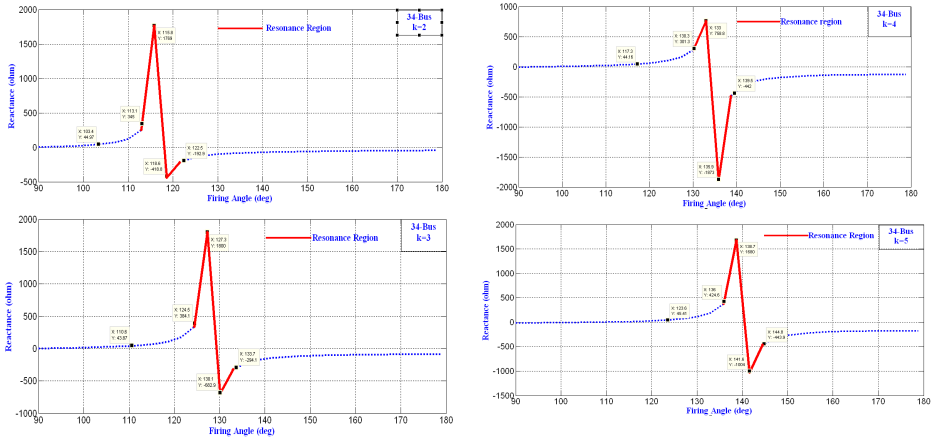


Fig. 9. Reactance curve between firing angle and X_{TCSC} for 34 bus system with various values of 'k'

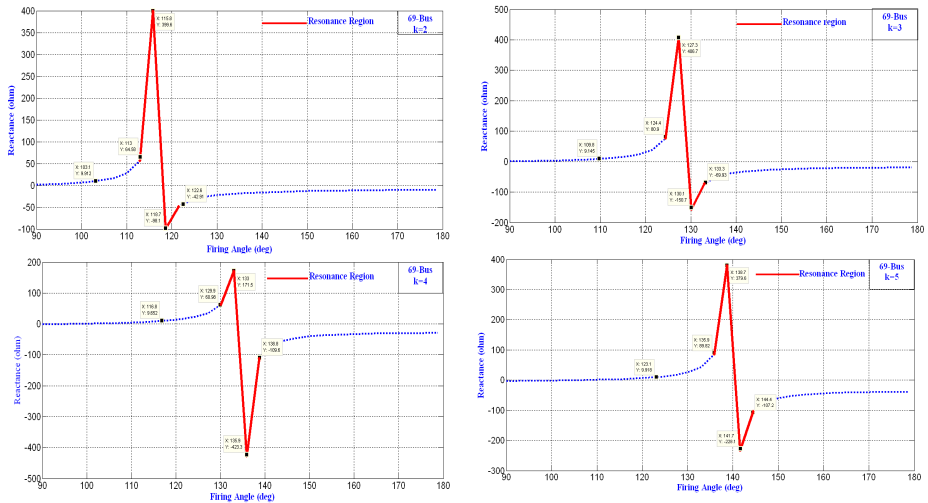


Fig. 10. Reactance curve between firing angle and X_{TCSC} for 69 bus system with various values of 'k'

10 Conclusion

The objective of this work, which is presented in this paper, is to reduce the active power loss of the radial distribution system with voltage profile improvement via optimal location and sizing of TCSC. Also in this work the main parameters of TCSC such as firing angle ' α ', X_L and X_C values are selected from the sizing of TCSC obtained from the PSO algorithm. This selection is done by the degree of series compensation ' k '. From the distance between the firing angle and the resonance region the best value of ' k ' is chosen. The proposed work has been tested using three different radial distribution systems and the results are tabulated. The firing angle obtained for various test systems shows the similarities for each value of ' k '. Thus the results demonstrate the effectiveness of the proposed work for placement, sizing and parameter selection of TCSC in FACTS compensated distribution systems.

11 Biographies

S.Devi received her B.E. Degree in Electrical and Electronics Engineering in 2002 from Manonmaium Sundaranar University, Tirunelveli and M.E degree in Power Electronics and Drives in 2004 from Sathyabama University, Chennai. She is pursuing her academic research in the area of Power Systems. Her field of interest is Power electronics applications in Power system.

M.Geethanjali received her B.E. Degree in Electrical and Electronics Engineering in 1989 and Master's degree in Power systems Engineering in 1991 from Madurai Kamaraj University. She has been awarded Ph.D. degree in the year 2008. She is in teaching profession for the past 21 years. She is a recipient of **Young Scientist Fellowship** awarded by Tamilnadu State Council for Science and Technology, Chennai, during the year 2004. At her credit she has 36 research papers.

Acknowledgment. The authors are thankful to the authorities of Thiagarajar College of Engineering, Madurai - 625015 India, for providing all the facilities to do the research work.

References

- [1] Taher, S.A., Hasani, M., Karimian, A.: A novel method for optimal capacitor placement and sizing in distribution systems with nonlinear loads and DG using GA. Commun. Nonlinear Sci. Numer. Simulat. 16, 851–862 (2011)
- [2] Kazemi, A., Andami, H.: FACTS devices in deregulated electric power systems: review. In: IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies, Hong Kong, pp. 337–342 (April 2004)
- [3] Verma, K.S., Singh, S.N., Gupta, H.O.: Facts Devices Location for Enhancement of TTC. In: IEEE Power Engineering Society Winter Meeting, January 28- February 1, vol. 2, pp. 522–527 (2001)
- [4] 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)

- [5] Price, K.V.: An introduction to differential evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) *New Ideas in Optimization*, pp. 79–108. Mc Graw-Hill, UK (1999)
- [6] Price, K., Storn, R.M., Lampinen, J.A.: *Differential Evolution: A Practical Approach to Global Optimization*. Springer (2005) ISBN: 3-540-20950-6
- [7] Gerbex, S., Cherkaoui, R., Germond, A.J.: Optimal location of multi-type facts devices in a power system by means of genetic algorithms. *IEEE Transactions on Power Systems* 16(3), 537–544 (2001)
- [8] Panda, S., Padhy, N.P.: Comparison of particle swarm optimization and genetic algorithm for FACTS based controller design. *Applied Soft Computing* 8(4), 1418–1427 (2008)
- [9] Hernandez, J.C., del Valle, Y., Venayagamoorthy, G.K., Harley, R.G.: Optimal allocation of a STATCOM in a 45 bus section of the Brazilian power system using particle swarm optimization. In: *IEEE Swarm Intelligence Symposium*, Indianapolis (2006)
- [10] Hingorani, N.G., Gyugyi, L.: *Understanding FACTS Concepts and Technology of Flexible AC Transmission Systems*, pp. 37–66. IEEE Press (2000)
- [11] Mathur, R.M., Varma, R.K.: *Thyristor based FACTS controllers for Electrical transmission systems*. John Wiley & Sons Inc.,
- [12] Acharya, N.: *Facts and Figures about FACTS.*, Training Workshop on FACTS Application AIT, EPSM, Energy, EPSM, Asian Institute of Technology (December 16, 2004)
- [13] Nagalakshmi, S., Kamaraj, N.: Secured loadability enhancement with TCSC in transmission system using computational intelligence techniques for pool and hybrid model. *Applied Soft Computing* 11, 4748–4756 (2011)
- [14] Meikandasivam, S., Nema, R.K., Jain, S.K.: Fine power flow control by split TCSC. *Electrical Power and Energy Systems* 45, 519–529 (2013)
- [15] Teng, J.-H.: A Direct Approach for Distribution System Load Flow Solutions. *IEEE Transactions on Evolutionary Computation* 18(3), 882–887 (2003)
- [16] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings, IEEE International Conf. on Neural Networks*, vol. 4, pp. 1942–1948
- [17] Gozel, T., Hakan Hocaoglu, M.: An analytical method for the sizing and siting of distributed generators in radial systems. *Electric Power System Research* 79, 912–918 (2009)

A New Particle Swarm Optimization with Population Restructuring Based Multiple Population Strategy

Qingjian Ni^{1,2}, Cen Cao¹, and Huimin Du³

¹ School of Computer Science and Engineering, Southeast University, Nanjing, China

² Provincial Key Laboratory for Computer Information Processing Technology,
Soochow University, Suzhou, China

³ College of Software Engineering, Southeast University, Nanjing, China
nqj@seu.edu.cn, {caocen2006, myosotis.blank}@gmail.com

Abstract. In view of the low convergence speed in traditional single population particle swarm optimization (PSO) when dealing with complicated optimization problems, this paper proposed a population restructuring based multiple population strategy, and combined this strategy with a comparatively new PSO variant (Dynamic Probabilistic Particle Swarm Optimization, DPPSO). With this strategy, more than one subpopulations are maintained during the evolution, and particles of each subpopulation evolve by a specific DPPSO variant independently. After a certain generations, restructure the present population stochastically to new subpopulations, and these restructured subpopulations will carry on to further evolution according to their previous DPPSO variant. Stochastic restructuring of population guarantees the diversity inside each subpopulation and even the whole population, which gives particles less chance to get plunging into local optima. Experiment on the classical benchmark functions demonstrates that DPPSO algorithms with the proposed strategy could efficiently avoid local optima with a considerable convergence speed.

Keywords: Particle Swarm Optimization, Multiple Population Strategy, Population Restructuring.

1 Introduction

Particle Swarm Optimization (PSO for short) is a type of optimization tool that is widely used for various of optimization problems in engineering fields [1][2][3].

There are a mass of applications for optimization problems, some of which are rather complex, like those with high dimension or multiple minima. Such problems are usually time-consuming to solve and hard to find satisfying solutions that's acceptable in engineering implement. When adopting PSO to such kind of problems, it is necessary to think about how to improve the solving efficiency and performance. Parallelization of PSO is an available method to improve solving performance of the algorithm. Under the conditions of the current hardware

capabilities, multiple population strategy is an important expression form of parallelization, and population based algorithm adopting the multiple population strategy usually possesses two or more subpopulations during implementing.

In fact, multiple population strategy is not only an important realization form of parallelization, meanwhile, it is an important method to improve the solving performance. Zhao et al. combined PSO with Harmony Search algorithm, divided the whole population into several subpopulations, and got good results on some benchmark functions [4]. Marinakis et al. adopted multiple population strategy and combined PSO with greedy algorithm to solve the probabilistic traveling salesman problem [5]. Liang et al. employed PSO to optimize multi-objective problems based on multiple population strategy [6]. Yu et al. proposed an ensemble of niching algorithms, which used several different parallel populations [7].

The previous researches demonstrate that, it has significant meaning to the performance of PSO in reality applications by the adoption of multiple population strategy, and these exploration about multiple population strategy will provide new thought and method in applying PSO to solve complex optimization problems.

However, researches about multiple population strategy of PSO is still not deep enough, for example, those subpopulations of multiple population strategy usually use the same evolutionary pattern. In this paper, combining the characteristics of the relatively new dynamic probabilistic particle swarm optimization(DPPSO for short)[8] [9], we proposed a new multiple population strategy based on population restructuring. And taking double populations for instance, we discussed and analyzed the effect that the strategy has on the performance of DPPSO with a couple of experiments. The rest of this paper is organized as follows. Section 2 gives an introduction to DPPSO, section 3 presents a detail analysis to the population restructuring based multiple population strategy which is proposed in this paper, section 4 discusses the results of the experiments, and section 5 draws a conclusion.

2 The Dynamic Probabilistic Particle Swarm Optimization

2.1 The Basic Concept of DPPSO

Since the particle swarm optimization has been invented, many variants of PSO have been proposed, and the more commonly used PSO variants include the PSO with constriction coefficient [10] and the PSO with inertia weight [11]. In these PSO variants, particles are used as candidate solutions in the solution space, with the position and explicit velocity attribute. Particles in the solution space evolve their positions with a certain speed, in order to find the optimal solution.

Based on the analysis of particle swarm evolutionary mechanism, Kennedy first designed a new variant of PSO [8] where there is no velocity attribute of particles. Ni launched in-depth research on this kind of PSO variant [9] [12]. Such

variants of PSO are generally known as Dynamic Probabilistic Particle Swarm Optimization (DPPSO). Unlike the usual PSO algorithms, particles have no velocities in the DPPSO algorithms, and the particles' position update are according to equation 1. Different from the traditional PSO algorithms, in the variant of DPPSO, particle has no velocity attribute and evolves to a new position in the form of probability which is also depending on the previous experience and position. The position update of DPPSO is according to equation 1.

$$X_i(t + 1) = X_i(t) + \alpha * (X_i(t) - X_i(t - 1)) + \beta * CT_i(t) + \gamma * Gen() * OT_i(t) \tag{1}$$

$$CT_{id}(t) = \sum_{k=1}^K P_{kd} / K - X_{id}(t) \tag{2}$$

$$OT_{id}(t) = \sum_{k=1}^K |P_{id} - P_{kd} / K| \tag{3}$$

In the position update equation 1, 2 and 3 of DPPSO, the meanings of symbols are shown in Table 1. $CT_i(t)$ and $OT_i(t)$ are D -dimensional vectors and will be determined by the particle's position and the experiences of its neighborhood ones. $Gen()$ is a random number generator, and usually the $Gen()$ should satisfy a distribution such as a Gaussian distribution.

Table 1. The meanings of the symbols in the position update equation of DPPSO

| Symbol in DPPSO | Meaning |
|-------------------------|--|
| i | The number of particle |
| t | The generation's number of evolution |
| $X_i(t)$ | The particle i 's position vector in the t th generation |
| k | The number of particle's neighborhood individual |
| d | The dimension's number of the particle's position vector |
| P_k | The optimum position vector of the particle k 's neighborhood ones |
| K | The quantity of particle's neighborhood |
| α, β, γ | The factor which is a positive constant |
| $Gen()$ | The dynamic probabilistic evolutionary operator |
| $OT_i(t)$ | A vector which is an abbreviation of Outlier Trend |
| $CT_i(t)$ | A vector which is an abbreviation of Centralized Tendency |

As can be seen from equation 1, the evolution of a particle is determined by the four aspects. The first aspect is the memory of particle's own position. The second aspect indicates the particle's movement trend along the former direction. The third aspect could be understood as the influence from the experiences of particle's neighborhoods. The fourth aspect considers the affect of the differences between a particle and its neighborhood ones' experiences.

2.2 The Variants of DPPSO

The flow of DPPSO is shown in algorithm 1.

Algorithm 1. The flow of Dynamic Probabilistic Particle Swarm Optimization

- 1 Randomly initialize the positions of particles in the whole population;
 - 2 Initialization of important parameters of DPPSO;
 - 3 **while** *The termination condition is not satisfied* **do**
 - 4 Evaluate fitness values of all the particle in the population;
 - 5 Update the optimal positions for each particle;
 - 6 For each particle, compute the *CT* values according to the formula 2;
 - 7 For each particle, compute the *OT* values according to the formula 3;
 - 8 For each particle, generate the new positions according to the formula 1;
-

Research about DPPSO conclude that different variant of DPPSO has its own superiority when adopting various dynamic probabilistic evolutionary operator *Gen()* [9]. DPPSO-Gaussian (also known as GDPS) has a faster convergence speed in the early stage of evolution. DPPSO-Logistic (also known as LDPSO) and DPPSO-Hyperbolic Secant still have an excellent capacity of exploration in the later period of evolution, which ensures that the particles will have a stronger ability to escape from local optima. DPPSO-Cauchy performs remarkably dealing with a few benchmark functions, which shows that it would be suitable to apply DPPSO-Cauchy to problems with some special features. These characteristics of DPPSO concluded from existing research will be conducive to solve real engineering problems.

3 Population Restructuring Based Multiple Population Strategy

In this paper, we presents an population restructuring based multiple population strategy according to the characteristics of DPPSO. This strategy could be expressed as follows. At the beginning of the algorithm, a population of N particles is generated, which is then divided into M subpopulations. Particles of each subpopulation will evolve independently on the basis of a specific DPPSO. After a certain number of generations of the evolution, restructure the population stochastically at this moment, and sent the restructured particles to each subpopulation. Then each subpopulation will carry on evolving with its previous DPPSO algorithm. In this strategy, different subpopulation applies different versions of DPPSO, thus, every subpopulation places particular emphasis on exploration and exploitation. And because of the stochastic restructuring at the certain moments, the diversity of subpopulations and the whole population is guaranteed, which makes it difficult for those particles to get trapped in local best solution. As a consequence, restructuring ensures the diversity of population; meanwhile, with different versions of DPPSO, a fast convergence is reached as well.

When the population restructuring based multiple population strategy is taken into use, the schematic diagram of the stochastic restructuring among particles at a certain moment is described in Figure 1 .

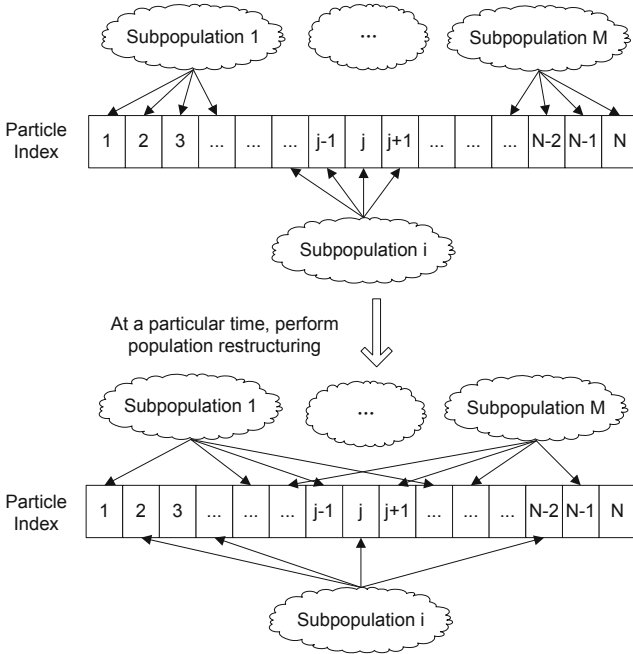


Fig. 1. Schematic diagram at a time of population restructuring

When adopting the population restructuring based multiple population strategy incorporated with DPPSO, the procedure is explained as Algorithm 2.

Population restructuring based multiple population strategy takes different DPPSO variants in the subpopulations, and particles among subpopulations exchange information through the stochastic restructuring operation, which improves the diversity of population. Hence, the capacity of global searching is enhanced. And eventually, the optimization ability of the whole population is boosted.

4 Experiments and Analysis

4.1 Experiment Settings

There are totally M subpopulations maintained during the implementation of algorithm, and different variants of DPPSO are chosen for each subpopulation. When dealing with practical problems, we could choose several suitable variants of DPPSO based on the problem's scale and other characteristics. Combine

Algorithm 2. The Procedure of DPPSO with the population restructuring based multiple population strategy

- 1 Initialize the master process and relevant parameters of DPPSO algorithm;
 - 2 Initialize the whole population in the solution space, and divide the population into M subpopulations, send the particles of those subpopulations to corresponding process;
 - 3 **for** each subpopulation i (the sequence number of subpopulation, $1 \leq i \leq M$) **do**
 - 4 Calculate the fitness value of particles in subpopulation i , and update the position of best individual of i ;
 - 5 Calculate the CT and OT values of particles in subpopulation i ;
 - 6 Generate the new positions of particles in subpopulation i ;
 - 7 **if** The population restructuring is in need **then**
 - 8 Send the information of particles in i to the master process;
 - 9 perform step 12;
 - 10 **if** The termination condition is satisfied **then**
 - 11 Send the information obtained by subpopulation i to the master process;
 - 12 Restructure the particles of the whole population stochastically, then divide particles into M groups, send the information of restructured particles to the corresponding subgenerations and take step 3;
 - 13 The master process analyzes and synthesizes the data from subpopulations, outputs the solution to the optimization problem.
-

with the analysis of various versions of DPPSO [9], DPPSO-Gaussian, DPPSO-Cauchy, DPPSO-Logistic, DPPSO-Hyperbolic Secant could all be available for the evolution of subpopulations. In this paper, we choose DPPSO-Logistic and DPPSO-Gaussian as the evolutionary method of subpopulations, and the biggest difference between DPPSO-Logistic and DPPSO-Gaussian is that these two variants are using different $Gen()$. The number of subpopulations could be equal to or more than 2 according to the actual scale and difficulty of the problem. Of course, it will cost less for controlling and maintaining and be easier to implement with less subpopulations.

Methods and moments of population restructuring could also influence the solving performance. If the restructuring operation took place too frequently during implementing, costs for calculating and communication would be increased obviously. In this paper, The group recombinant method adapted in this paper are: the whole population are restructured every 10 generations.

To measure the efficiency of the proposed strategy, experiments settings are as follows: the number of subpopulations (i.e. M) is set to 2, DPPSO-Logistic and DPPSO-Gaussian are selected as evolutionary algorithms adapted by subpopulations, and the classical fully connected topology(i.e. G_{best} model) is used inside each subpopulation. Four benchmark functions are tested, which are described in Table 2.

In the algorithm using the proposed strategy, the total population size is 50, size of each subpopulation is 25. And DPPSO-Logistic and DPPSO-Gaussian

with the single population size of 50 are used for comparing. Thus, even the algorithm of multiple population strategy is operated in one machine system, the calculating costs for fitness evaluation would be the same. The three algorithms are repeated 100 times during the experiment. The key performance indicators are investigated which include the optimal value, the median value, the mean value, the standard deviation, the worst value and the success rate, etc.

Table 2. The Instructions of the Four Benchmark Functions

| | | | | | |
|-------------|-----------|--|---------------|-----------------|----------------|
| Sphere | Formula | $f(\mathbf{x}) = \sum_{i=1}^n x_i^2$ | | | |
| | Dimension | Optimal Solution | Optimal Value | Value Range | Accepted Error |
| | 30 | (0, 0, 0, ..., 0) | 0 | $ x_i < 100$ | 0.01 |
| Rastrigin | Formula | $f(\mathbf{x}) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$ | | | |
| | Dimension | Optimal Solution | Optimal Value | Value Range | Accepted Error |
| | 30 | (0, 0, 0, ..., 0) | 0 | $ x_i < 5.12$ | 100 |
| Rosenbrock | Formula | $f(\mathbf{x}) = \sum_{i=1}^{n-1} [(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2]$ | | | |
| | Dimension | Optimal Solution | Optimal Value | Value Range | Accepted Error |
| | 30 | (1, 1, 1, ..., 1) | 0 | $ x_i \leq 30$ | 100 |
| Schaffer F6 | Formula | $f(\mathbf{x}) = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} - 0.5$ | | | |
| | Dimension | Optimal Solution | Optimal Value | Value Range | Accepted Error |
| | 2 | (0, 0) | 0 | $ x_i < 100$ | 0.00001 |

4.2 Results and Analysis

After 100 times' repeat of experiment, data from three algorithms are presented in Table 3, 4, 5 and 6, which include the optimal value, the median value, the mean value, the standard deviation, the worst value and the success rate. Figure 2 and Figure 3 show evolutionary trend of optimal fitness for benchmark functions.

(1) Sphere Function

Data in Table 3 shows that, the proposed strategy performs better and is more stable for the 65 dimensional Sphere function combining with the indexes

Table 3. Comparison of Results in Sphere Function

| Algorithm | Optimal | Median | Mean | SD | Worst | Success rate |
|---------------------|----------|----------|----------|----------|----------|--------------|
| DPPSO-Gaussian | 2.20E-97 | 9.00E-41 | 0.003215 | 0.031148 | 0.31151 | 99% |
| DPPSO-Logistic | 6.29E-66 | 2.98E-65 | 3.42E-65 | 2.31E-65 | 1.32E-64 | 100% |
| Multiple Population | 2.20E-80 | 3.96E-79 | 2.49E-08 | 2.49E-07 | 2.49E-06 | 100% |

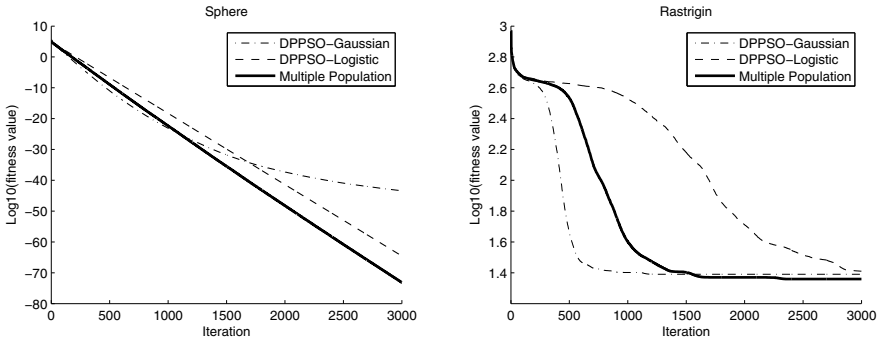


Fig. 2. Comparison of evolutionary trend between four topologies (Sphere and Rastrigin)

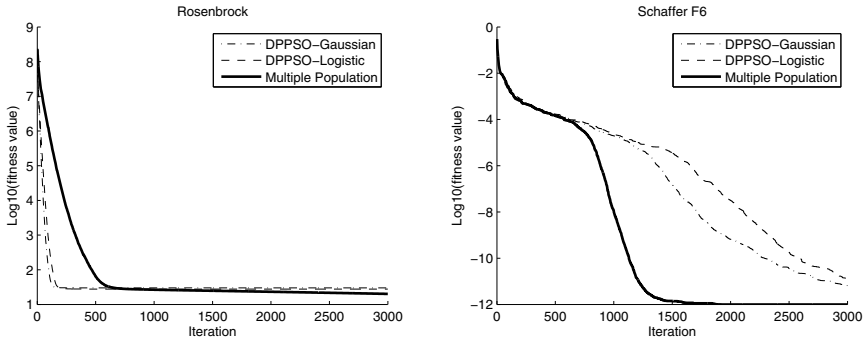


Fig. 3. Comparison of evolutionary trend between four topologies (Rosenbrock and Schaffer F6)

of the median value and the success rate. Figure 2 also indicates that during the evolution, especially in the middle and later period, the proposed strategy retains the tendency to get close to theoretical optima and is quite leading. Relatively, DPPSO-Gaussian performs better on the optimal value only, but the order of magnitudes in the optimal value and the median value have a huge interval. This illustrates that the proposed strategy functions more steady than DPPSO-Gaussian. That is to say, combining experimental data of these algorithms, the proposed strategy is particular dominant in handling Sphere function.

(2) Rastrigin Function

Data in Table 4 shows that, for the Rastrigin function of 60 dimension, the solutions of each algorithm from repeatedly testing do not differ much in the order of magnitudes. On such occasion, functions of algorithms could be distinguished by the mean value and the standard deviation. The proposed strategy precedes DPPSO-Logistic and DPPSO-Gaussian on the mean value, and its success rate reaches 100%, which reflects its better stability. Combining with Figure 2, the

proposed strategy takes over the lead in the middle and the later evolution, and still keeps the tendency to get closer to theoretical optimal solution.

Table 4. Comparison of Results in Rastrigin Function

| Algorithm | Optimal | Median | Mean | SD | Worst | Success rate |
|---------------------|---------|--------|--------|--------|--------|--------------|
| DPPSO-Gaussian | 11.962 | 24.874 | 25.148 | 5.2583 | 39.798 | 100% |
| DPPSO-Logistic | 9.9496 | 20.894 | 49.356 | 90.573 | 402.23 | 90% |
| Multiple Population | 12.934 | 22.884 | 23.314 | 4.4902 | 31.839 | 100% |

(3) Rosenbrock Function

Data in Table 5 shows that, for the 30 dimensional Rosenbrock function, the solutions of each algorithm from repeatedly testing have no much difference in the order of magnitudes. And the index of the mean value and the standard deviation would be more convincing. The proposed strategy keeps ahead not only on the mean value and the standard deviation, but even on all other indices, which fully embodies its superiority in solving the problem. Meanwhile, Figure 3 also indicates that, though the proposed strategy does not stand out at the early stage, but its dominant position becomes clear in the middle and later period, and is keeping the tendency towards theoretical optimal solution.

Table 5. Comparison of Results in Rosenbrock Function

| Algorithm | Optimal | Median | Mean | SD | Worst | Success rate |
|---------------------|---------|--------|--------|--------|--------|--------------|
| DPPSO-Gaussian | 27.15 | 27.514 | 33.139 | 23.078 | 191.88 | 97% |
| DPPSO-Logistic | 26.832 | 27.173 | 28.382 | 8.5202 | 87.852 | 100% |
| Multiple Population | 19.305 | 19.866 | 20.478 | 5.7068 | 76.922 | 100% |

(4) Schaffer F6 Function

Data in Table 6 shows that, compared with DPPSO-Logistic and DPPSO-Gaussian, the proposed strategy performs overwhelmingly on all the indices. Furthermore, its success rate in this experiment reached 100%. This fully demonstrates the excellent ability of the proposed strategy. As can be seen in Figure 3, the proposed strategy keeps ahead consistently and finds solution quickly.

Table 6. Comparison of Results in Schaffer F6 Function

| Algorithm | Optimal | Median | Mean | SD | Worst | Success rate |
|---------------------|---------|--------|----------|----------|----------|--------------|
| DPPSO-Gaussian | 0[87%] | 0 | 5.57E-05 | 0.00034 | 0.003167 | 94% |
| DPPSO-Logistic | 0[76%] | 0 | 5.16E-07 | 4.71E-06 | 4.71E-05 | 99% |
| Multiple Population | 0[100%] | 0 | 0 | 0 | 0 | 100% |

Results from the above experiments demonstrate that the population restructuring based multiple population strategy has ideal performance in experiments and is particular dominant compared with DPPSO-Logistic and DPPSO-Gaussian of single population. Also, it is particularly worth noting that, the proposed strategy achieved excellent results for solving problems that is hard to optimize, such as Schaffer F6 and Rosenbrock. That means the proposed strategy could be quite useful to solve problems with similar characteristics.

5 Conclusion

This paper proposed a multiple population strategy based on the population restructuring. When adopting this strategy, subpopulations select different variant of DPPSO for evolution, particles in different subpopulations focus on different emphasis between exploration and exploitation, and information communication among the whole population is realized through the population restructuring. From another point of view, single particle does not evolve according to a unique evolutionary way. This helps particles improve their abilities to escape from local optimal.

In conclusion, the proposed strategy took advantage of the diversities of different DPPSO, and applied different DPPSO variant to divided subpopulations, and this type of approach showed its excellence in dealing with complicated optimization problems according to the results and analysis of the experiments. Hence, when utilize DPPSO to real-world problems, the proposed strategy could be strongly available. Further studies would include the settings of size and scale about subpopulations, and the variant of DPPSO adopting in this strategy is also worthy of discussion.

Acknowledgements. This paper is supported by Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, China and NSFC (Grant No. 60803061, 61170164).

References

1. Kulkarni, R.V., Venayagamoorthy, G.K.: Particle swarm optimization in wireless-sensor networks: A brief survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 41(2), 262–267 (2011)
2. AlRashidi, M., El-Hawary, M.: A survey of particle swarm optimization applications in electric power systems. *IEEE Transactions on Evolutionary Computation* 13(4), 913–918 (2009)
3. Esmin, A.A., Coelho, R.A., Matwin, S.: A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data. *Artificial Intelligence Review*, 1–23 (2013)
4. Zhao, S.Z., Suganthan, P.N., Pan, Q.K., Fatih Tasgetiren, M.: Dynamic multi-swarm particle swarm optimizer with harmony search. *Expert Systems with Applications* 38(4), 3735–3742 (2011)

5. Marinakis, Y., Marinaki, M.: A hybrid multi-swarm particle swarm optimization algorithm for the probabilistic traveling salesman problem. *Computers & Operations Research* 37(3), 432–442 (2010)
6. Liang, J.J., Qu, B.Y., Suganthan, P.N., Niu, B.: Dynamic multi-swarm particle swarm optimization for multi-objective optimization problems. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8. IEEE (2012)
7. Yu, E., Suganthan, P.N.: Ensemble of niching algorithms. *Information Sciences* 180(15), 2815–2833 (2010)
8. Kennedy, J.: Dynamic-probabilistic particle swarms. In: *Proceedings of the Conference on Genetic and Evolutionary Computation*, pp. 201–207. ACM (2005)
9. Ni, Q., Deng, J.: Two improvement strategies for logistic dynamic particle swarm optimization. In: Dobnikar, A., Lotrič, U., Šter, B. (eds.) *ICANNGA 2011, Part I*. LNCS, vol. 6593, pp. 320–329. Springer, Heidelberg (2011)
10. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* 6(1), 58–73 (2002)
11. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: *Proceedings of IEEE World Congress on Computational Intelligence*, pp. 69–73. IEEE (1998)
12. Ni, Q., Deng, J.: A new logistic dynamic particle swarm optimization algorithm based on random topology. *The Scientific World Journal* 2013 (2013)

Small Signal Stability Constrained Optimal Power Flow Using Swarm Based Algorithm

Mani Devesh Raj¹ and Periyasami Somasundaram²

¹ Dept. of Electrical and Electronics Engg.,
SSN College of Engineering, Chennai
deveshraj.m@gmail.com

² Dept. of Electrical and Electronics Engg.,
College of Engineering, Guindy, Anna University, Chennai
mpsomasundaram@yahoo.com

Abstract. This paper presents a simple and efficient Particle Swarm Optimization (PSO) based algorithm for solving Small Signal Stability Constrained Optimal Power Flow (SSSCOPF) problem. The proposed methodology is based on an Optimal Power Flow (OPF) problem that explicitly considers small signal stability security limits, hence appropriately termed as SSSCOPF. The SSSCOPF problem is solved using PSO based algorithm. The Eigen value analysis is performed to assess the system stability for the normal operating condition and subsequently for stressed operating condition by considering single line outage. The proposed PSO based algorithm for solving OPF and SSSCOPF problems is tested on an adapted IEEE 30-bus system. The optimal solutions obtained for the OPF problem with and without small signal stability security constraints are compared and analyzed. The comparative analysis reveals that the proposed PSO based algorithm converges to the optimal solution without any constraint violations.

1 Introduction

The main task of power system operator is to adjust certain controllable system variables like active power generation from power plants, generator terminal voltage, reactive power compensation and on-load tap changers of transformers etc., so that the best operating point can be achieved. Optimal power flow (OPF) is an important tool used by the power system operators both in planning and operating stages to minimize the total fuel cost of thermal generating units while satisfying certain constraints like bus voltage magnitude limits, line flow limits, reactive power generation limits etc., [1-3]. The OPF problem can also include constraints that represent operation of the system after certain contingency outages. These “security constraints” allow the OPF to dispatch the generation to the system in a defensive manner. That is, the OPF now forces the system to operate so that if a contingency occurs, then the resulting bus voltages and line flows would still be within their limits ensuring a secure operation of the entire power system. Therefore this special type of OPF is termed as “Security Constrained Optimal Power Flow” (SCOPF).

The optimal power flow solution has one primary limitation that it does not cater for system security constraints like small signal stability pertaining to increase of load or outage case states. Hence OPF with security constraints is to be developed which allows re-dispatching with an appropriate security level in terms of small signal stability [11-12]. It can be observed that stability has become an important consideration in OPF. Though some have investigated including transient stability limits [8][10][12-14][16-17][20], but only a few authors have considered small signal stability [15-18]. This paper proposes an OPF method taking into account small signal stability as additional constraint and hence termed as Small Signal Stability Constrained Optimal Power Flow (SSSCOPF).

Several optimization techniques, such as Linear Programming, Non-Linear Programming (NLP), quadratic programming and interior point methods are used for solving the security constrained generation scheduling problems [1-3]. Dommel and Tinney [1] presented a penalty function based NLP technique to solve optimal power flow problem. Alsac and Stott [2] extended the penalty function method to security constrained optimal power flow problem in which all the contingency case constraints are augmented to the optimal power flow problem. In this method the functional inequality constraints are handled as soft constraints using penalty function technique. The drawback of this approach is the difficulty involved in choosing proper penalty weights for different systems and different operating conditions which if not properly selected may lead to excessive oscillatory convergence.

Linear programming and dynamic programming techniques, for example, often fail (or reach local optimum) in solving large scale NP problems with large number of variables and non-linear objective functions. To overcome these problems researchers have proposed stochastic approaches like Evolutionary Programming (EP) and Particle Swarm Optimization (PSO) techniques [9][15][18-19][21-29]. Somasundaram et al [3] presents an algorithm for solving security constrained optimal power flow problem through the application of EP. PSO is a heuristic global optimization method, which is based on swarm intelligence [24-28]. Compared to other stochastic approach PSO has the advantage that it is easy to implement and there are few parameters to adjust [29]. Hence in this paper PSO based algorithm is adopted to realize the SSSCOPF methodology.

2 SSSCOPF Problem Formulation

The SSSCOPF is formulated as a mathematical optimization problem with various constraints as follows,

Objective:

$$\text{Min } F_T = \sum_{i=1}^{ng} (a_{Gi} P_{Gi}^2 + b_{Gi} P_{Gi} + c_{Gi}) + a_{GS} P_{GS}^2 + b_{GS} P_{GS} + c_{GS} \quad (1)$$

where 'ng' is the total number of generators except slack bus generator.

Subject to:

Constraints on control variables,

$$P_{Gj,\min} \leq P_{Gj} \leq P_{Gj,\max}; j \in ng \quad (2)$$

$$V_{Cl,\min} \leq V_{Cl} \leq V_{Cl,\max}; l \in \alpha NVC \quad (3)$$

Base-case power flow equations,

$$F(X, U, C) = 0 \quad (4)$$

The state vector X comprises of the bus voltage phase angles and magnitudes, the control vector U comprises of all the controllable system variables and the parameter vector C includes all the uncontrollable system parameters such as line parameters, loads, etc.

Slack bus generator constraint,

$$P_{Gs,\min} \leq P_{Gs} \leq P_{Gs,\max} \quad (5)$$

Constraints on dependent variables,

$$Q_{Gq,\min} \leq Q_{Gq} \leq Q_{Gq,\max}; q \in \alpha NVC \quad (6)$$

$$V_{Lr,\min} \leq V_{Lr} \leq V_{Lr,\max}; r \in \alpha NVL \quad (7)$$

$$LF_k \leq LF_{k,\max}; k = 1, 2, \dots, NL \quad (8)$$

Contingency case power flow equations,

$$F^*(X^*, U, C^*) = 0 \quad (9)$$

The triplet (X^*, U, C^*) characterize stressed state. The state vector X^* comprises of the bus voltage phase angles and magnitudes, the control vector U comprises of all the controllable system variables and the parameter vector C^* includes all the uncontrollable system parameters such as line parameters, loads, etc. for the stressed state (like increase in load or transmission line outages).

Other limits of contingency case are reactive power generation, load bus voltage magnitude and line flow security constraints

$$P_{Gs,\min} \leq P_{Gs}^* \leq P_{Gs,\max} \quad (10)$$

$$P_{Gj,\min} \leq P_{Gj}^* \leq P_{Gj,\max} \quad (11)$$

$$Q_{Gq,\min} \leq Q_{Gq}^* \leq Q_{Gq,\max} \quad (12)$$

$$V_{Lr,\min} \leq V_{Lr}^* \leq V_{Lr,\max} \quad (13)$$

$$LF_k^* \leq LF_{k,\max} \quad (14)$$

Small signal stability constraints,

$$\zeta_k \geq \zeta_{min} \quad \text{and} \quad \zeta_k \neq 0 ; \tag{15}$$

Where $\zeta_k = \frac{\sigma_k}{\sqrt{\sigma_k^2 + \omega_k^2}}$, and $\sigma_k + j\omega_k$ is the k^{th} Eigen value for both normal and contingency condition using Eigen value analysis.

3 PSO Based Algorithm for Solving SSSCOPF

The SSSCOPF problem is solved in two phases. In the first phase, the base-case OPF problem is formulated and solved using the PSO technique [19-22] and [3]. In the next phase, the SSSCOPF problem is formulated and solved using PSO based technique by taking the optimal solution obtained in phase I as initial feasible solution for SSSCOPF problem. The flow chart of SSSCOPF methodology is shown in Fig.1.

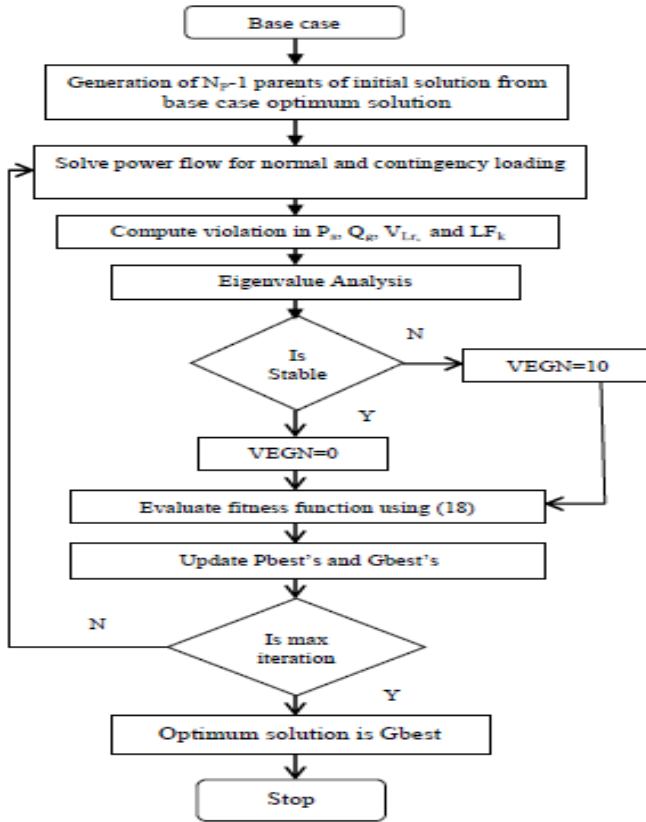


Fig. 1. Flow chart of SSSCOPF

The various sequential steps for solving SSSCOPF using PSO based algorithm is as follows,

For the SSSCOPF initially ‘Np’ swarm of particles are obtained by selecting the initial single particle from the optimum solution obtained in the base case OPF problem as,

$$I_i = [V_{G1i}, \dots, V_{G1m}, \dots, V_{G1env}, P_{G1i}, \dots, P_{G1j}, \dots, P_{G1ng}] \quad (16)$$

The remaining (Np-1) particle of initial swam is generated from this initial particle as,

$$P_{Gij} = P_{Gij} + rand() \times \beta \times (P_{Gj,max} - P_{Gj,min}) ; V_{Gij} = V_{Gij} + rand() \times \beta \times (V_{Cl,max} - V_{Cl,min}) ; \quad (17)$$

Where $i=2, \dots, Np$, $P_{Gj,max}$ and $P_{Gj,min}$ are the maximum and minimum MW generation for the j^{th} generation unit and β is the scaling factor (random number between -1 to 1). For each initial particle I_i power flow is conducted using eqn. (4) and eqn. (9) for both normal and contingency conditions. And their corresponding slack bus generations, load bus voltage magnitudes, phase angles of all buses (excluding slack bus) and all line flows are computed. For each particle, real parts of Eigen values are computed under normal and contingency case using small signal stability analysis as in [4-7]. Subsequently evaluate the fitness function value of each particle I_i using eqn. (18).

$$\begin{aligned} Min f_i = & F_T + (k_1 \times VSG) + (k_2 \times VSQ) + (k_3 \times VLF) + (k_4 \times VLB) \\ & + (k_5 \times VEGN) + (k_6 \times VSG^*) + (k_7 \times VSQ^*) + (k_8 \times VLF^*) \\ & + (k_9 \times VLB^*) + (k_{10} \times VEGN^*) \end{aligned} \quad (18)$$

Where,

$VSG =$ Slack bus real power violation

$$if P_{GS}^{min} \leq P_{Gsi} \leq P_{GS}^{max}, VSG=0$$

$$else if P_{Gsi} \leq P_{GS}^{min}, VSG = |P_{Gsi} - P_{GS}^{min}|$$

$$else if P_{Gsi} \geq P_{GS}^{max}, VSG = |P_{GS}^{max} - P_{Gsi}| \quad (19)$$

$VSQ =$ Total generator bus reactive power violation

$$if Q_{GS}^{min} \leq Q_{Gsi} \leq Q_{GS}^{max}, VSQ=0$$

$$else if Q_{Gsi} \leq Q_{GS}^{min}, VSQ = |Q_{Gsi} - Q_{GS}^{min}|$$

$$else if Q_{Gsi} \geq Q_{GS}^{max}, VSQ = |Q_{GS}^{max} - Q_{Gsi}| \quad (20)$$

$VLF =$ Total Line flow violation

$$if LF_{iz} \geq LF_z^{max}, VLF = \sum_{z=1}^{nl} |LF_z^{max} - LF_{iz}|$$

$$else if VLF = 0 \quad (21)$$

$VLB =$ Total load bus voltage violation

$$if V_{Lj} \leq V_{Lj}^{min}, VLB_j = |V_{Lj} - V_{Lj}^{min}|$$

$$else if V_{Lj}^{max} \leq V_{Lj}, VLB_j = |V_{Lj}^{max} - V_{Lj}| \quad (22)$$

$VEGN = \text{Small signal stability violation}$

$$\text{If } f_k < f^* \text{ or } f_k > 0, VEGN = 10$$

$$\text{else} = 0; \tag{23}$$

and VSG^* , VSQ^* , VLF^* , VLB^* , $VEGN^*$ corresponds to violation in contingency cases.

Pbest is initialized with particle having minimum fitness function and Gbest the best of Pbest is found. Velocity of particle is updated and the member position of each particle is modified and new Gbest is found [24][25][26][27].When maximum iteration is reached the Gbest gives the optimum solution.

4 Sample System Studies and Results

The PSO based algorithm discussed in the previous section is tested on an adapted IEEE 30-bus [1-2] to assess the performance of the proposed algorithm. The adapted IEEE 30-bus system consists of six generators 41 lines and a total demand of 189.2MW and 106.6 MVAR [23]. The bus data, line data and MVA line flow limits and generation cost data for the system are taken from [1-2] and the machine data from [4-7]. The objective function is the total fuel cost and the fuel cost curve of the units is represented by quadratic cost functions [2]. The lower voltage-magnitude limits at all buses are 0.95 p.u., and the upper limits are 1.1 p.u. for generator buses 1, 2, 5, 8, 11 and 13 the upper limits are 1.05 p.u.. For PSO based algorithm the following parameters are chosen as, $t_{max}=200$, $Np=10$, $c_1=1$, $c_2=1$, $IW_{min}=0.1$ and $IW_{max}=0.9$. The penalty factors for the constraint violations $k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}$ are 1000, 10000, 10000, 1000, 1000, 1000, 10000, 10000, 1000 and 1000 respectively. The Small signal Stability violation limit f^* is chosen as -3.5×10^{-6} .

4.1 Conventional OPF without Security Constraints Results (Phase I)

The OPF problem is solved using the proposed PSO based algorithm and the optimum schedule, total fuel cost and transmission loss are given in Table 1.

Table 1. Optimal Solution for OPF without Security Constraints

| PARAMETERS | REAL POWER (MW) | REACTIVE POWER (MVAr) | BUS VOLTAGES (in pu) |
|------------|-----------------|-----------------------|----------------------|
| G1 | 113.990 | 8.66 | 1.010 |
| G2 | 33.721 | 18.827 | 0.996 |
| G5 | 15.000 | 3.100 | 0.987 |
| G8 | 10.000 | 29.791 | 0.966 |
| G11 | 10.000 | 21.844 | 1.044 |
| G13 | 12.000 | -14.137 | 0.950 |

Table 2. (continued.)

| | |
|--------------------------------------|----------|
| Total thermal generation (MW) | 194.7104 |
| Total load (MW) | 189.20 |
| Losses (MW) | 5.5547 |
| Total fuel cost (\$/hr) | 490.11 |

The important point is that there is no limit violation in the base case corresponding to the optimum schedule. This fact demonstrates that the proposed algorithm is very robust and reliable in eliminating the limit violations. Eigen value analysis for normal loading and contingency case (transmission line outage of line connecting buses 3 and 4) is performed with the normal OPF result and is given in Table 2. From the Eigen value analysis it is inferred that real parts of all the Eigen values corresponding to the normal operating conditions are negative, thereby the system is small signal stable. But for a transmission line outage (connecting buses 3 and 4) real parts of two Eigen values are positive, hence the optimal solution of OPF problem without security constraints tends to suffer small signal instability issues. Hence there is a need for SSSCOPF methodology.

Table 3. Eigen Values for the Optimal Solution of OPF without Security Constraints

| Sl.No. | For normal loading | For contingency case |
|---------------|---------------------------|-----------------------------|
| 1 | -0.0006 +33.5440i | -0.0006 +33.6651i |
| 2 | -0.0006 -33.5440i | -0.0006 -33.6651i |
| 3 | -0.0004 +26.3767i | 0.0001 +26.3434i |
| 4 | -0.0004 -26.3767i | 0.0001 -26.3434i |
| 5 | -0.0036 +12.8132i | -0.0036 +13.2598i |
| 6 | -0.0036 -12.8132i | -0.0036 -13.2598i |
| 7 | -0.0002 + 8.7760i | 0.0001 + 8.9820i |
| 8 | -0.0002 - 8.7760i | 0.0001 - 8.9820i |
| 9 | -0.0079 + 6.0491i | -0.0110 + 5.8130i |
| 10 | -0.0079 - 6.0491i | -0.0110 - 5.8130i |
| 11 | -0.0000 | -0.0000 |
| 12 | -0.1529 | -0.1578 |
| 13 | -0.3374 | -0.3109 |
| 14 | -0.4085 | -0.4082 |
| 15 | -0.4609 | -0.4602 |
| 16 | -0.4522 | -0.4525 |
| 17 | -0.4402 | -0.4401 |

4.2 SSSCOPF Results (Phase II)

Single-line outage of line between buses numbered 3 and 4 is considered. The optimum solution of the conventional OPF without security constraints (phase I) is used to generate the initial swarm for SCOPF.

The convergence characteristic of fitness function is shown in Fig. 2. The fitness function convergence characteristic is drawn by taking the particle with minimum fitness value at the end of every iteration. It is observed that from Fig. 2 the fitness function converges smoothly to the optimum value without any abrupt oscillations. This shows the convergence reliability of the proposed SSSCOPF algorithm with single line outage as a constraint.

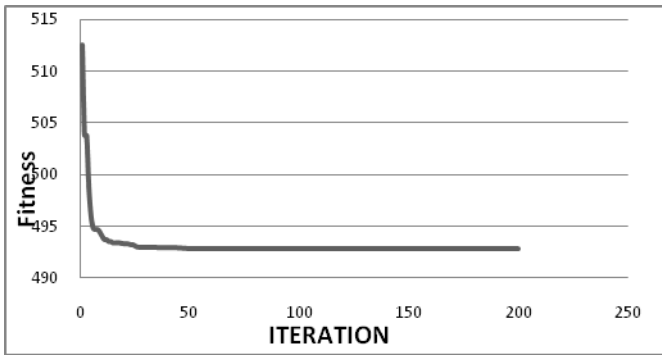


Fig. 2. Convergence characteristic of fitness function for base-case OPF

The optimum schedule, total fuel cost and transmission loss for SSSCOPF problem are given in Table 3.

Table 3. Optimal Solution for SSSCOPF

| PARAMETERS | REAL POWER (MW) | REACTIVE POWER (MVar) | BUS VOLTAGES (in pu) |
|--|-----------------------|-----------------------------|----------------------------|
| G1 | 100.325 | 18.713 | 1.010 |
| G2 | 38.729 | -1.346 | 0.992 |
| G5 | 19.082 | 1.052 | 0.986 |
| G8 | 10.000 | 40.893 | 0.974 |
| G11 | 14.072 | 21.945 | 1.047 |
| G13 | 12.000 | -15.083 | 0.950 |
| Total thermal generation (MW) | | 194.206 | |
| Total load (MW) | | 189.20 | |
| Losses (MW) | | 5.043 | |
| Total fuel cost (\$/hr) | | 494.35 | |

From Table 3., it is inferred that the generators active and reactive power and voltage limits are not violated. Moreover from Table 1 and Table 3 it is inferred that there is an increase in total fuel cost with the inclusion of security constraint. For the optimal solution presented in Table 3, Eigen value analysis for two operating conditions namely the normal loading and a transmission line outage (between buses 3 and 4) is performed and is presented in Table 4.

Table 4. Eigen Values for the Optimal Solution of SSSCOPF

| Slno. | For normal loading | For contingency case |
|-------|--------------------|----------------------|
| 1 | -0.0011 +33.6240i | -0.0011 +33.7347i |
| 2 | -0.0011 -33.6240i | -0.0011 -33.7347i |
| 3 | -0.0005 +26.3097i | -0.0001 +26.2744i |
| 4 | -0.0005 -26.3097i | -0.0001 -26.2744i |
| 5 | -0.0054 +12.2496i | -0.0050 +12.6967i |
| 6 | -0.0054 -12.2496i | -0.0050 -12.6967i |
| 7 | -0.0007 + 8.9882i | -0.0005 + 9.1542i |
| 8 | -0.0007 - 8.9882i | -0.0005 - 9.1542i |
| 9 | -0.0054 + 6.1258i | -0.0070 + 5.8985i |
| 10 | -0.0054 - 6.1258i | -0.0070 - 5.8985i |
| 11 | -0.0000 | -0.0000 |
| 12 | -0.1516 | -0.1566 |
| 13 | -0.3440 | -0.3207 |
| 14 | -0.4068 | -0.4067 |
| 15 | -0.4601 | -0.4595 |
| 16 | -0.4488 | -0.4495 |
| 17 | -0.4396 | -0.4395 |

From Table 4 it is inferred that real parts of all the Eigen values corresponding to the normal operating conditions as well as for a transmission line outage (between buses 3 and 4) are negative, thereby the system is small signal stable. Thus the effectiveness of the PSO based algorithm for solving SSSCOPF problem is reinforced.

5 Conclusion

This paper presents an efficient and simple approach for solving the SSSCOPF problem. This paper demonstrates the general OPF problem formulation and subsequently the SSSCOPF problem formulation are presented along with the various

sequential steps for solving them using a PSO based algorithm. The applicability and effectiveness of the proposed PSO based technique is assessed using the adapted IEEE 30-bus system. The optimal solutions for OPF and SSSCOPF are analyzed. The analysis reports that there are no constraint violations including the small signal stability constraint for the optimal solution obtained by the PSO based algorithm and the fitness value converges to the optimum without any abrupt oscillations. Thereby the effectiveness of the PSO based algorithm is reinforced.

References

- [1] Alsac, O., Stott, B.: Optimal load flow with steady state security. *IEEE Trans. PAS* 93(3), 745–751 (1974)
- [2] Somasundaram, P., Kuppusamy, K., Kumudini Devi, R.P.: Evolutionary programming based security constrained optimal power flow. *Electric Power Systems Research* 72, 137–145 (2004)
- [3] Capitanescua, F., Wehenkel, L.: Experiments with the interior-point method for solving large scale Optimal Power Flow problems. *Electric Power Systems Research* 95, 276–283 (2013)
- [4] Kundur, P.: *Power System Stability and Control*. McGraw-Hill, New York (1994)
- [5] Sauer, P.W., Pai, M.A.: *Power System Dynamics and Stability*. Prentice-Hall, Upper Saddle River (1998)
- [6] Anderson, P.M., Fouad, A.A.: *Power System Control And Stability* (2008)
- [7] Ramanujan, R.: *Power System Dynamics Analysis And Design*. PHI Learning (2010)
- [8] Zarate-Minano, R., Van Cutsem, T., Milano, F., Conejo, A.J.: Securing Transient Stability Using Time-Domain Simulations Within an Optimal Power Flow. *IEEE Transactions on Power Systems* 25, 243–253 (2010)
- [9] Attia, A.-F., Al-Turki, Y.A., Abusorrah, A.M.: Optimal Power Flow Using Adapted Genetic Algorithm with Adjusting Population Size. *Electric Power Components and Systems* 40(11), 1285–1299 (2012)
- [10] Su, C., Chen, Z.: An Optimal Power Flow (OPF) Method with Improved Power System Stability. In: *Universities Power Engineering Conference (UPEC)*, pp. 1–6 (2010)
- [11] Condren, J., Gedra, T.W.: Expected-security cost optimal power flow with small-signal stability constraints. *IEEE Trans. Power Syst.* 21(4), 1736–1743 (2006)
- [12] Gan, R.J., Thomas, Z.R.D.: Stability-constrained optimal power flow. *IEEE Trans. Power Syst.* 15(2), 535–540 (2000)
- [13] Yuan, J., Kubokawa, S.H.: A solution of optimal power flow with multi-contingency transient stability constraints. *IEEE Trans. Power Syst.* 18(3), 1094–1102 (2003)
- [14] Ruiz-Vega, D., Pavella, M.: A comprehensive approach to transient stability control: part I—near optimal preventive control. *IEEE Trans. Power Syst.* 18(4), 1446–1453 (2003)
- [15] Zhao, B., Guo, C.X., Cao, Y.J.: A multiagent-based particle swarm optimization approach for optimal reactive power dispatch. *IEEE Transactions on Power Systems* 20(2), 1070–1078 (2005)
- [16] Kodsi, S.K.M., Canizares, C.A.: Application of a stability constrained optimal power flow to tuning of oscillation controls in competitive electricity markets. *IEEE Transactions on Power Systems* 22(4), 1944–1954 (2007)
- [17] Cai, H.R., Chung, C.Y., Wong, K.P.: Application of differential evolution algorithm for transient stability constrained optimal power flow. *IEEE Transactions on Power Systems* 23(2), 719–728 (2007)

- [18] Yoshida, H., Kawata, K., Fukuyama, Y., Takayama, S., Nakanishi, Y.: A particle swarm optimization for reactive power and voltage control considering voltage security assessment. *IEEE Transactions on Power Systems* 15(4), 1232–1239 (2000)
- [19] del Valle, Y., Venayagamoorthy, K., Mohagheghi, Carlos Hernandez, J., Harley, R.G.: Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Transactions on Evolutionary Computation* 12(2), 171–195 (2008)
- [20] Mo, N., Zou, Z.Y., Chan, K.W., Pong, G.: Transient stability constrained optimal power flow using particle swarm optimization. *IET Generation, Transmission & Distribution* 1, 476–483 (2007)
- [21] Kumar, S., Chaturvedi, D.K.: Optimal power flow solution using fuzzy evolutionary and swarm optimization. *International Journal of Electrical Power and Energy Systems* 47, 416–423 (2013)
- [22] Zaro, F.R., Abido, M.A.: Multi-Objective Particle Swarm Optimization for Optimal Power Flow in a Deregulated Environment of Power Systems. In: 2011 11th International Conference on Intelligent Systems Design and Applications, pp. 1122–1127 (2011)
- [23] Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* 6(1), 58–73 (2002)
- [24] Mohapatra, A., Bijwe, P.R., Panigrahi, B.K.: Optimal Power Flow with Multiple Data Uncertainties. *Electric Power Systems Research* 95, 160–167 (2013)
- [25] Chowdhury, B.H., Rahman, S.: A review of recent advances in economic dispatch. *IEEE Trans. Power Syst.* 5, 1248–1259 (1990)
- [26] Arul, R., Ravi, G., Velusami, S.: Solving Optimal Power Flow Problems Using Chaotic Self-adaptive Differential Harmony Search Algorithm. *Electric Power Components and Systems* 41(8), 782–805 (2013)
- [27] Pandya, K.S., Joshi, S.K.: Sensitivity and Particle Swarm Optimization-based Congestion Management. *Electric Power Components and Systems* 41(4), 465–484 (2013)
- [28] Arul, R., Ravi, G., Velusami, S.: Chaotic self-adaptive differential harmony search algorithm based dynamic economic dispatch. *International Journal of Electrical Power and Energy Systems* 50, 85–96 (2013)
- [29] de Fátima Araújo, T., Uturbey, W.: Performance assessment of PSO, DE and hybrid PSO–DE algorithms when applied to the dispatch of generation and demand. *International Journal of Electrical Power and Energy Systems* 47, 205–221 (2013)

Online Voltage Stability Assessment of Power System by Comparing Voltage Stability Indices and Extreme Learning Machine

M.V. Suganyadevi and C.K. Babulal

Department of Electrical and Electronics Engineering,
Thiagarajar College of Engineering, India
{mvsee, ckbeee}@tce.edu

Abstract. Nowadays the modern power systems are large complex systems and widely distributed geographically. The increase in demand, generator reaching reactive power limits and line/generator outages may operate the power system in stressed conditions leads to voltage instability or voltage collapse. Voltage stability has become of major concern among the power utilities, because of several events of voltage collapse occurred in the past decade. Thus the real time voltage stability assessment is essential by estimating the loadability margin of the power system. Voltage stability analysis was studied on IEEE 30 and IEEE 118 bus systems using many voltage stability indices based on line/nodal and the results are compared with the proposed new index based on SVM and ELM.

Keywords: voltage stability assessment, voltage stability index, support vector machine, extreme learning machine.

1 Introduction

Problems related to voltage stability have recently been considered as the major concerns in the planning and operation of power systems. The rapid increase in load demand in electric power system motivates the researchers to protect the power system to restrain voltage collapse. Voltage stability is concerned with the ability of power systems to maintain acceptable voltages at all buses in the system under normal conditions and after being subjected to a disturbance[1-3]. Voltage instability has been attributed to the lack of adequate reactive support and the difficulty in the flow of required reactive power on the transmission network.

In most of the cases, voltage profiles show no abnormality prior to undergoing voltage collapse because of the load variation. Voltage stability margin (VSM) is a static voltage stability index which is used to quantify how “close” a particular operating point is to the point of voltage collapse [4]. Thus, VSM may be used to estimate the steady-state voltage stability limit of a power system. Knowledge of the

voltage stability margin is of vital importance to utilities in order to operate their system with appropriate security and reliability. The system operator must be provided with an accurate and fast method to predict the voltage stability margin so as to initiate the necessary control actions.

During the last few years, several methodologies for detecting the voltage collapse points (saddle-node bifurcations) in power systems using steady-state analysis techniques have been modified and applied for the determination of analyzing voltage stability of power systems for example PV and QV curves, sensitivity-based indices and continuation power flow methods [5,6]. Other methods, such as bifurcation theory[7], energy function, singular value decomposition and so forth, have been also reported in the literature.

Several static voltage stability methods are also available in the literature for the quick assessment of voltage stability. In literature [8-16] several methods have been proposed to identify critical bus bars, critical line, and stability margins of the power system.

Some of the indices are:

1. Line Stability Index (L_{mn}).
2. Fast Voltage Stability Index (FVSI).
3. Line Stability Factor (LQP).
4. Voltage Collapse Proximity Indices (VCPI).
5. L-Index (L).
6. Three Diagonal Element dependent Index (I_{pi} , I_{qi} , I_i).

This paper compares the performance of the above said voltage stability indices for different loading scenarios like real power load increase, reactive power load increase and both real and reactive power load increase of IEEE 30 and IEEE 118 bus test systems [17]. In addition a new index is proposed based on Extreme Learning Machine (ELM). The proposed index takes real and reactive power load as input parameters and gives a voltage stability margin called ELM-VSI.

The Extreme learning machine (ELM) was proposed recently as an efficient learning algorithm for single-hidden layer feed forward neural network (SLFN) [19]. The Extreme learning machine (ELM) [20-23] studies a much wider type of “generalized” SLFNs whose hidden layer need not be tuned. ELM was originally developed for the single-hidden-layer feed forward neural networks and then extended to the “generalized” SLFNs which may not be neuron alike and compared with the index based on SVM [18].

Organization of the paper is as follows: Section 2 describes various VSI used to examine the voltage stability of the system. Section 3 presents the ELM and the test systems and analysis tools used are presented in section 4. Results and discussions are presented in section 5. Finally conclusions are summarized in section 6.

2 Voltage Stability Margin

Voltage stability margin is defined as difference between maximum transferring power that system can be tolerated and power of normal operation. In this paper, this voltage stability margin is referred to as the loadability margin and used as performance index for voltage stability analysis. The purpose of VSI is to determine the point of voltage instability, the weakest bus in the system and the critical line referred to a bus. Indices proposed based on bus are known as Nodal Indices and that of transmission lines are Line Indices. The Nodal Indices and Line Indices are briefly discussed in the following section.

2.1 Line Voltage Stability Indices

The line indices are based on the bus power (real and reactive), voltage magnitude, phase angle and impedance of the transmission line as shown in the Fig.1. The factor

$D = \left[\left(\frac{r}{x} \sin \delta + \cos \delta \right) V_i \right]^2 - 4 \left(x + \frac{r^2}{x} \right) Q_j \geq 0$ for stable operating condition. If the factor D is less than zero, it indicates the roots are imaginary and the system is unstable.

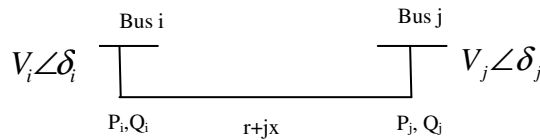


Fig. 1. Typical one line diagram of transmission line

The Line Stability index L_{mn} derived from the factor D assuming lossless transmission line is as follows

$$L_{mn} = \frac{4XQ_j}{[V_i \sin(\theta - \delta)]^2} \tag{1}$$

Line stability factor LQP index is based on a concept of power flow through a single line. LQP is calculated as

$$LQP = 4 \left(\frac{X}{V_i} \right) \left(\frac{X}{V_i^2} P_i^2 + Q_j \right) \tag{2}$$

The fast voltage stability index FVSI is formulated based on a power transmission line.

$$FVSI = \frac{4Z^2 Q_j}{V_i^2 X} \tag{3}$$

The Voltage Collapse Proximity Index VCPI investigates the stability of each line of the system and they are based on the concept of maximum power transferred through a line.

$$VCPI_{(P)} = \frac{P_j}{P_{j(\max)}} \tag{4}$$

$$VCPI_{(Q)} = \frac{Q_j}{Q_{j(\max)}} \tag{5}$$

Where the values P_j and Q_j are obtained from conventional power flows calculations, and $P_{j(\max)}$ and $Q_{j(\max)}$ are the maximum active and reactive power that can be transferred through the line. The line indices varies from 0(no load) to 1 (maximum loadability). The value of indices that is evaluated close to 1.00 indicates that the particular line is closed to its instability that may lead to voltage collapse. To maintain system security the value of indices should be maintained well below the value of 1.00.

2.2 Nodal Indices

The L-Index is a quantitative measure for the estimation of the distance from the actual state of the system to the stability limit. The index can be computed as

$$L = \max_{j \in nL} (L_j) = \max_{j \in n} \left| 1 - \frac{\sum_{i \in nG} F_{ji} V_i}{V_j} \right| \tag{6}$$

F_{ji} is the submatrice of [H]

$$\text{Where } H = \begin{bmatrix} Z_{LL} & F_{LG} \\ K_{GL} & Y_{GG} \end{bmatrix}$$

L_j is a local indicator that determinates the busbars from where collapse may originate. The L index varies in a range between 0 (no load) and 1 (maximum loadability).

Diagonal element dependent indices I_p, I_q

These indices are observed that with increase in load at the load bus, the value of diagonal elements of the Jacobian $\partial Q_i / \partial V_i$ and $\partial P_i / \partial \delta_i$ gets reduced. Thus, the deviation in value of $\partial Q_i / \partial V_i$ and $\partial P_i / \partial \delta_i$ from its no-load value to the value at any particular loading condition can be used as index of voltage stability for the load bus i. Using this criteria two voltage stability indices are proposed:

$$I_{qi} = \frac{\partial Q_i / \partial V_i}{-B_{ii}} \tag{7}$$

$$I_{pi} = \frac{\partial P_i / \partial \delta_i}{-B_{ii}} \quad (8)$$

The threshold value of the VSI are proposed as

$$I_i = \frac{\partial P_i / \partial \delta_i}{\sum_{\substack{j=1 \\ j \neq i}}^N B_{ij} V_j} \quad (9)$$

Where B_{ii} is the susceptance of the line connected to i^{th} load bus

3 Extreme Learning Machine

The ELM algorithm was originally proposed by Huang *et al.* in [20] and it makes use of the SLFN. The main concept behind the ELM lies in the random initialization of the SLFN weights and biases

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \quad (10)$$

where $\mathbf{h}(\mathbf{x})$ is the hidden-layer output corresponding to the input sample \mathbf{x} and \mathbf{h} is the output weight vector between the hidden layer and the output layer. One of the salient features of ELM is that the hidden layer need not be tuned. Essentially, ELM originally proposes to apply random computational nodes in the hidden layer, which are independent of the training data. Different from traditional learning algorithms for a neural type of SLFNs [24], ELM aims to reach not only the smallest training error but also the smallest norm of output weights. ELM [25,26] and its variants [27,28] mainly focus on the *regression* applications. Latest development of ELM has shown some relationships between ELM and SVM [29,30].

4 Test System and Analytical Tool

For the analysis of voltage stability, the test systems – IEEE 30 and IEEE 118 bus systems are considered. All the results are produced with the help of a program developed in PSAT. PSAT is MATLAB software for electric power system analysis and control [32]. Though several loading pattern are tested for two test systems, due to space limitations only some scenarios are presented. In scenario 1, real power load at the weakest load bus alone increased. In scenario 2, reactive power load at the weakest load bus alone increased. In scenario 3, both real and reactive power load at the all load buses are increased simultaneously. The above procedure is illustrated in Fig. 3

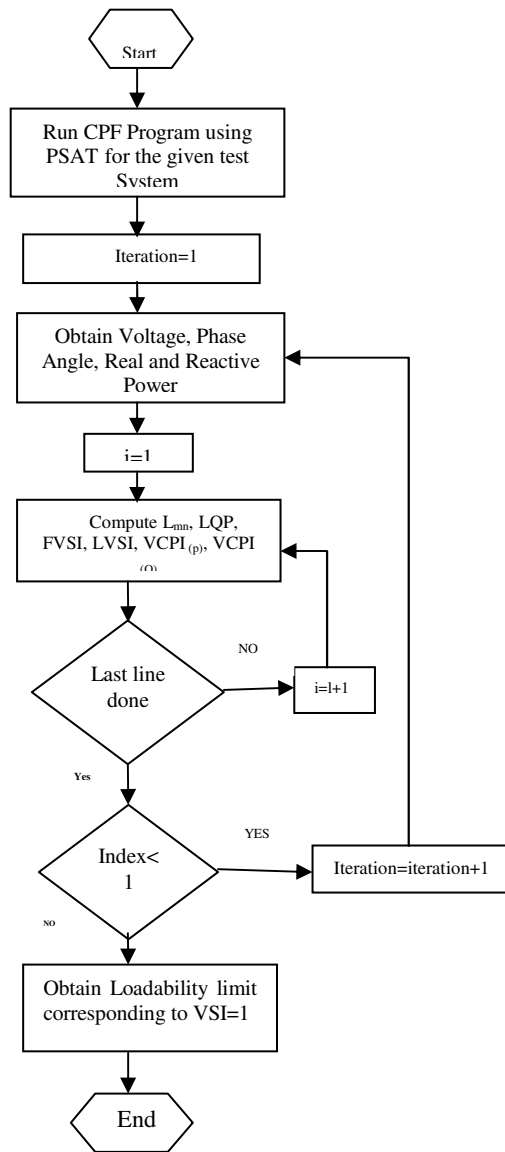


Fig. 2. Flow Chart for the estimation of loadability margin using VSI

5 Results and Discussions

5.1 Estimation of Loading Margin Using VSI and CPF of IEEE 30 Bus System

In order to compare the VSI, the loadability margin of IEEE 30 bus system is calculated for three scenarios. Scenario 1, the real power demand at weakest load bus is increased gradually. In scenario 2, the reactive power demand at the weakest load bus is increased gradually. Finally in scenario 3, both the real and reactive power demand at the weakest load bus is increased gradually.

In Scenario 1: The real power demand at bus 24 is 10% increased from its base value. Continuation Power Flow (CPF) available in PSAT is used to trace the PV curve for the above case. The real and reactive power load obtained in each iteration of the CPF is used to calculate the voltage stability indices (L_{mn} , LQP, FVSI, $VCPI_{(p)}$, $VCPI_{(q)}$, L, I, I_p , and I_q). Fig.4. (a) to (d) shows the variation of line voltage stability indices and Fig.4.(e) to (g) shows the variation of nodal voltage stability indices with respect to real power demand at bus 24. The following observations are made from the figures 4. (a) to (d).

Line Voltage Stability Indices:

- Maximum loadability limit at bus 24 obtained from CPF is $P_{max} = 0.055$ p.u.
- As real power load increases the FVSI, L_{mn} , LQP are also increases as expected.
- However FVSI, L_{mn} , LQP are unable to detect the P_{max} . Because FVSI, L_{mn} , LQP are less than '1(p.u)' at P_{max} . This clearly indicates that FVSI, L_{mn} , and LQP gives higher loadability limit than the actual value from CPF.
- $VCPI_{(p)}$ increases gradually to P_{max} . Hence, $VCPI_{(p)}$, exactly determines the loadability limit for this scenario.

Nodal Voltage Stability Indices:

Inspection of Fig. 4 (e) to (g) reveals the following observation.

- As real power load increases, L, 'I' and I_p indices are also increases.
- But the L index is unable to detect the P_{max} exactly. Because it gives a value greater than 1.00 value at P_{max} .
- The 'I' and I_p indices threshold value is 0.5 at the maximum loading point. This clearly indicates that 'I' and I_p indices gives higher loadability limit than the actual value from CPF.

In Scenario 2 : The reactive power at bus 24 is 10% increased from its base case value. Continuation Power Flow available in PSAT is used to trace the QV curve. For the same real and reactive power load obtained in each iteration of the CPF, the voltage stability indices (L_{mn} , LQP, FVSI, $VCPI_{(p)}$, $VCPI_{(q)}$, L, I, I_p , I_q) are determined. The results are compared in the following table. Q_{max} is the maximum loadability limit at bus 24. First row in Table 1 presents the value of VSI corresponding to maximum loadability limit and the Second row indicate the value of loadability limit corresponding to each index at its threshold.

Line Voltage Stability Indices:

As reactive load power increases the FVSI, L_{mn} , LQP are also increases as expected. However FVSI, L_{mn} , LQP are unable to detect the Q_{max} . Because FVSI, L_{mn} , LQP are less than 1 at Q_{max} . This clearly indicates that FVSI, L_{mn} , LQP give higher loadability limit than the actual. VCPI ($VCPI_{(p)}$ & $VCPI_{(q)}$) increases gradually at Q_{max} , VCPI becomes 1. Hence VCPI, exactly determines the loadability limit for this load scenario 2.

Table 1. Loadability margin from VSI and CPF for scenario 2 and scenario 3

| | Load at 24 th bus | | | |
|--------------------|------------------------------|--------|------------|--------|
| | Scenario 2 | | Scenario 3 | |
| | VSI | CPF | VSI | CPF |
| | LM=0.067 (p.u) | | | |
| | LM= 0.095 (p.u) | | | |
| LMN(P.U) | 1.5894 | 0.5876 | 1.009 | 0.8790 |
| LQP (P.U) | 1.0780 | 0.7971 | 1.089 | 0.7792 |
| FVSI (P.U) | 0.9923 | 0.4002 | 1.047 | 0.6602 |
| $VCPI_{(p)}$ (P.U) | 1.0001 | 0.9534 | 0.9323 | 0.9156 |
| $VCPI_{(q)}$ (P.U) | 0.9899 | 0.9221 | 0.9329 | 0.9166 |
| L (P.U) | 2.5673 | 0.8794 | 1.304 | 0.8999 |
| I_p (P.U) | 0.6932 | 0.3602 | 0.7930 | 0.4537 |
| I_q (P.U) | 0.6934 | 0.3554 | 0.7291 | 0.3777 |
| I (P.U) | 0.4534 | 0.1955 | 0.6665 | 0.3471 |

Nodal Voltage Stability Indices:

Generally L index increases with the reactive load increase. But it is unable to detect the Q_{max} exactly. Because L index gives a greater value than the actual load at Q_{max} . The I_p , I_q and I indices are also increase with the reactive load. Its threshold value is 0.5 at the maximum loading point. So the indices values are always greater than its actual Q_{max} value.

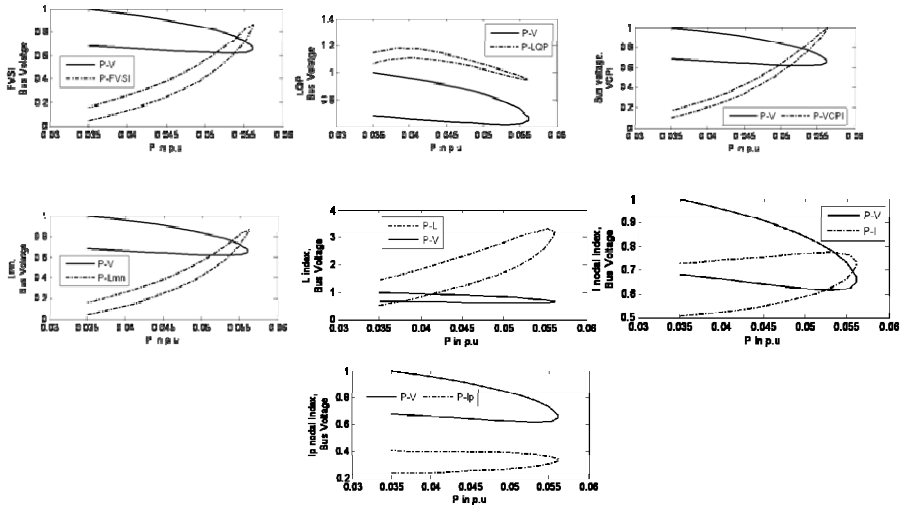


Fig. 3. (From clockwise) (a) Real power Vs voltage and FVSI, (b) Real power Vs voltage and LQP, (c) Real power Vs voltage and VCPI, (d) Real power Vs voltage and L_{mn} , (e) Real power Vs , voltage and L, (f) Real power Vs voltage and I_p , (g) Real power Vs voltage and I

In Scenario 3: In this scenario, both real and reactive power load is incremented by 10%. In this type of loading also, LQP, L_{mn} , FVSI, I , I_p , I_q indices unable to detect the maximum loadability limit exactly except VCPI ($VCPI_{(p)}$ & $VCPI_{(q)}$) index. The results are presented in Table 1.

5.2 Estimation of Loading Margin Using VSI and CPF of IEEE 118 Bus System

The loadability margin of IEEE 118 bus system is calculated for all the three scenarios namely increase in the real power demand, reactive power demand and both real and reactive power.

In Scenario 1, The real power demand at bus 88 is 10% increased from its base value. Continuation Power Flow (CPF) available in PSAT is used to trace the PV curve for the above case. The real and reactive power load obtained in each iteration of the CPF is used to calculate the voltage stability indices (L_{mn} , LQP, FVSI, $VCPI_{(p)}$, $VCPI_{(q)}$, L , I , I_p , and I_q). From the Table 2, the following observations are made.

- The maximum loadability limit at bus 88 obtained from CPF is $P_{max} = 2.05$ p.u.
- As real power load increases the Line Voltage Stability Indices FVSI, L_{mn} , LQP are also increases as expected.
- However FVSI, L_{mn} , LQP are unable to detect the P_{max} . Because FVSI, L_{mn} , LQP are less than '1' at P_{max} . This clearly indicates that FVSI, L_{mn} , and LQP gives higher loadability limit than the actual value from CPF.
- $VCPI_{(p)}$ increases gradually to P_{max} . Hence, $VCPI_{(p)}$ exactly determines the loadability limit for this scenario 1.
- As real power load increases, Nodal Voltage Stability Indices L , 'I', I_q and I_p **indices are also increases.**
- But the L index is unable to detect the P_{max} exactly. Because L index gives a value greater than 1.00 value at P_{max} .
- The 'I', I_q and I_p indices threshold value is 0.5 at the maximum loading point. This clearly indicates that the indices give higher loadability limit than the actual value from CPF.

In Scenario 2

The reactive power at bus 88 is 10% increased from its base case value. Continuation Power Flow available in PSAT is used to trace the QV curve. For the same real and reactive power load obtained in each iteration of the CPF, the voltage stability indices (L_{mn} , LQP, FVSI, $VCPI_{(p)}$, $VCPI_{(q)}$, L , I , I_p , I_q) are determined. The results are compared in the following table. Q_{max} is the maximum loadability limit at bus 24. From the table results, As reactive load power increases the Line VSI FVSI, L_{mn} , LQP are also increases as expected. However FVSI, L_{mn} , LQP are unable to detect the Q_{max} . Because FVSI, L_{mn} , LQP are less than 1 at Q_{max} . This clearly indicates that FVSI, L_{mn} , LQP give higher loadability limit than the actual. $VCPI_{(p)}$ ($VCPI_{(p)}$ & $VCPI_{(q)}$) increases gradually at Q_{max} , $VCPI$ becomes 1. Hence $VCPI$, exactly determines the loadability limit for this load scenario 2. And the Nodal Voltage

Stability Indices L, 'I', I_q, I_p indices also increases with the reactive load increase. But the indices are unable to detect the Q_{max} exactly. Because the indices give a greater value than the actual maximum loadability limit Q_{max}. 'I', I_q and I_p index's threshold value is 0.5 at the maximum loading point. So the indices values are greater than its actual maximum loadability margin.

In Scenario 3, both real and reactive power load is incremented by 10%. In this type of loading also, LQP, L_{mn}, FVSI, I, I_p, I_q indices unable to detect the maximum loadability limit exactly except VCPI (VCPI_(p) & VCPI_(q)) index. The results are presented in the following Table 2 .

Table 2. Loadability Margin from VSI and CPF of IEEE 118 Bus Systems

| | Load at 88 th bus | | | | | |
|---------------------------|------------------------------|-------|------------|---------------|---------------|-------|
| | Scenario 1 | | Scenario 2 | LM=2.99 (P.U) | Scenario 3 | |
| | LM=2.05 (P.U) | | | | LM=3.04 (P.U) | |
| | VSI | CPF | VSI | CPF | VSI | CPF |
| LMN(P.U) | 1.400 | 0.886 | 1.009 | 0.8790 | 1.138 | 0.879 |
| LQP (P.U) | 1.086 | 0.945 | 1.089 | 0.7792 | 1.084 | 0.778 |
| FVSI (P.U) | 1.289 | 0.876 | 1.047 | 0.6602 | 1.146 | 0.854 |
| VCPI _(p) (P.U) | 1.112 | 0.995 | 0.9323 | 0.9156 | 1.118 | 0.998 |
| VCPI _(q) (P.U) | 1.114 | 0.996 | 0.9329 | 0.9166 | 1.115 | 0.993 |
| L (P.U) | 1.776 | 0.453 | 1.304 | 0.8999 | 1.096 | 0.755 |
| I _p (P.U) | 0.999 | 0.339 | 0.7930 | 0.4537 | 0.559 | 0.344 |
| I _q (P.U) | 0.997 | 0.337 | 0.7291 | 0.3777 | 0.641 | 0.398 |
| I (P.U) | 0.664 | 0.442 | 0.6665 | 0.3471 | 0.663 | 0.345 |

5.3 Estimation of Loading Margin by SVM and ELM

Generation of Data

The required data is generated using CPF method available in PSAT. In IEEE 30 bus test system, As many as 597 patterns were generated by changing the load at each bus and generation randomly in wide range ($\pm 50\%$ of base case). Thus 35820 (597x60) load samples were generated. Power factor at all load buses are maintained constant. Out of 35820 data samples in IEEE 30 bus system, 80% of total samples (28656) were selected arbitrarily for training, while 20% (7184) were used for testing. The data samples used for testing the SVM model and ELM model are unseen values that are not used in training. In IEEE 118 Bus Test System, As many as 300 patterns were generated by changing the demand and generation at each bus randomly in wide range ($\pm 50\%$ of base case). Totally 35400 data samples were generated. Out of 300 patterns, 80% load scenarios (28320 samples) were arbitrarily selected for training while, 20% load scenarios (7080 samples) were used for testing the performance of the estimation of loadability margin.

Algorithm

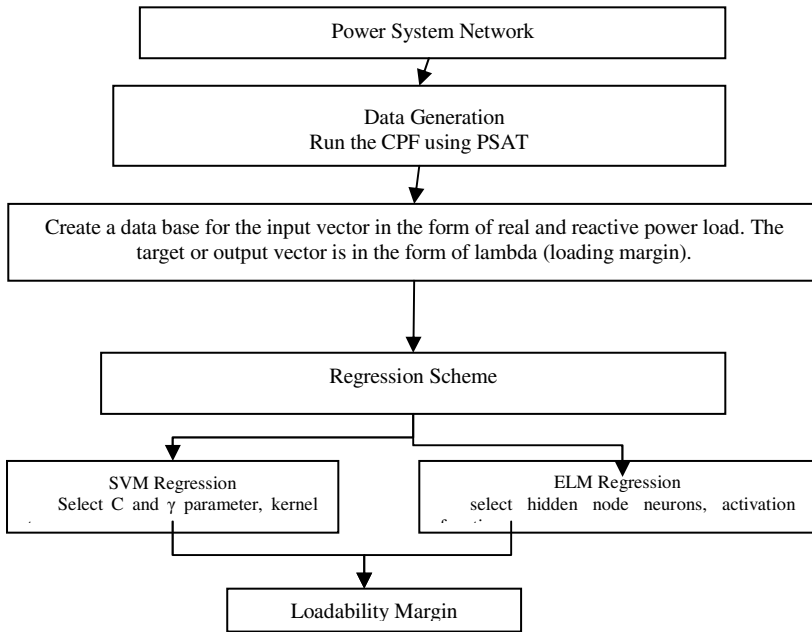


Fig. 4. Flow Chart for the estimation of LM using SVM and ELM

Table 3. LM comparisons among ELM, CPF, and SVM

| No | IEEE 30 bus system | | | IEEE 118 bus system | | |
|----|--------------------|--------|----------|---------------------|---------|----------|
| | CPF | SVM | ELM | CPF | SVM | ELM |
| 1 | 1.18906 | 1.0148 | 1.01484 | 1.066508 | 1.07684 | 1.076811 |
| 2 | 1.00116 | 1.3205 | 1.32049 | 1.168095 | 1.06470 | 1.064689 |
| 3 | 1.60025 | 1.6073 | 1.60731 | 1.264029 | 1.24729 | 1.247270 |
| 4 | 2.14578 | 1.8738 | 1.87386 | 1.354573 | 1.32478 | 1.324782 |
| 5 | 2.09961 | 2.1188 | 2.11875 | 1.439952 | 1.42737 | 1.427375 |
| 6 | 2.47358 | 2.3406 | 2.34065 | 1.906011 | 1.89737 | 1.897375 |
| 7 | 2.86492 | 2.5376 | 2.537659 | 1.954922 | 1.96516 | 1.965166 |
| 8 | 2.57753 | 2.7067 | 2.706636 | 1.999516 | 2.02815 | 2.028154 |
| 9 | 3.05481 | 2.8395 | 2.839433 | 2.039769 | 2.08643 | 2.086434 |
| 10 | 3.01280 | 2.8809 | 2.880869 | 2.075612 | 2.14008 | 2.140087 |

The voltage stability margin by CPF, SVM model and ELM are compared in Table 3 respectively for few testing patterns due to limited space. But the CPF is unable to detect the loading margin exactly, because it gives a value greater than 1.00 value at P_{max} when compared to ELM and SVM. The tables show clearly that the proposed ELM model estimate the same loadability margin as obtained by the conventional techniques with greater accuracy. The MSE values of IEEE test systems were simulated for SVM and ELM for two different activation functions namely sigmoid and RBF are tabulated in Table 4. The training computational time of

SVM-sigmoid is slightly higher than the SVM-RBF. The ELM-RBF the training and testing time are faster and accurate when compared to its sigmoid activation function. The SVM and ELM MSE are also tabulated in order to show the accuracy of the regression types. However the results show that the RBF types predicts the result quickly when compared to sigmoid type and the computational time also less for the estimation of loadability margin of a power system. The results show that the ELM network is able to produce the output with good accuracy(10^{-3}). The MSE and the computational time for the system obtained are also very less in the order of 10^{-4} and in few seconds respectively.

Table 4. Comparison of SVM and ELM

| | Regression Scheme | Training data samples | Testing data samples | Activation Function | Training Time (sec) | Testing Time (sec) | MSE |
|---------------------|-------------------|-----------------------|----------------------|---------------------|---------------------|--------------------|-----------------|
| IEEE 30 Bus System | SVM | 28656 | 7184 | Sigmoid | 9.812 | 0.0998 | $3.255e^{-005}$ |
| | | | | RBF | 7.956 | 0.0447 | $2.192e^{-006}$ |
| | ELM | 28656 | 7184 | Sigmoid | 7.462 | 0.0863 | $1.881e^{-005}$ |
| | | | | RBF | 5.578 | 0.0274 | $1.005e^{-006}$ |
| IEEE 118 Bus System | SVM | 28320 | 7080 | Sigmoid | 25.665 | 0.134 | $3.234e^{-005}$ |
| | | | | RBF | 18.830 | 0.116 | $2.784e^{-006}$ |
| | ELM | 28320 | 7080 | Sigmoid | 23.436 | 0.129 | $1.925e^{-005}$ |
| | | | | RBF | 16.991 | 0.109 | $1.028e^{-006}$ |

The Fig.6 and Fig 7 shows the prediction of loadability margin for IEEE 30 and IEEE 118 test bus systems respectively by comparing the two regression scheme models: SVM and ELM in terms of MSE for some testing patterns. The ELM-RBF predicts the loadability margin more quickly and accurately when compared to SVM-RBF.

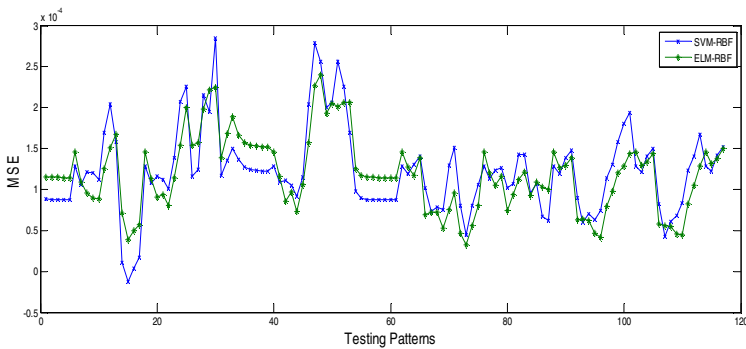


Fig. 5. Comparison of MSE among SVM and ELM of IEEE 30 bus system

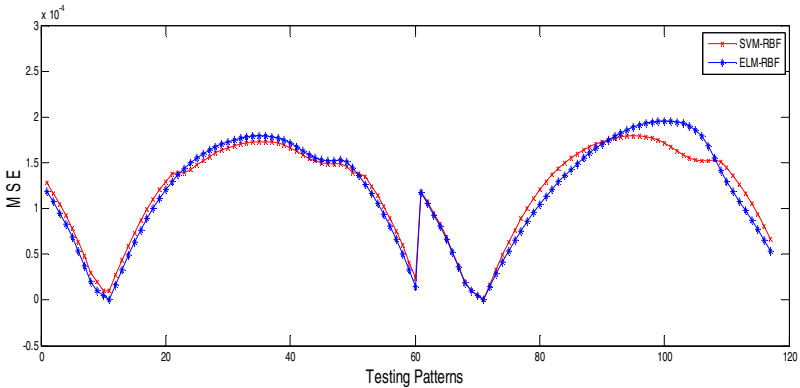


Fig. 6. Comparison of MSE among SVM and ELM of IEEE 118 bus system

6 Conclusion

In this paper, a new voltage stability index based on extreme learning machine technique is proposed which is used to evaluate the power system voltage stability. The performance of proposed ELM-VSI and all Voltage stability indices (L_{min} , LQP, FVSI, $VCPI_{(p)}$, $VCPI_{(q)}$, L , I , I_p , I_q) has been tested on IEEE 30 bus and IEEE 118 bus systems to evaluate the loadability margin of power systems. The already proposed voltage stability indices (L_{min} , LQP, FVSI, $VCPI_{(p)}$, $VCPI_{(q)}$, L , I , I_p , I_q) in the literature are unable to predict the exact loadability margin (LM) as that of CPF. The VSI based on ELM technique could able to identify the expected LM value with an error occurrence of 10^{-4} . A comparison of the propose index with other indices indicates that the ELM-VSI is a much more reliable indicator of the relative closeness to the voltage stability limit of a heavily loaded power systems. The proposed ELM-VSI can be implemented for on-line security assessment in Energy Management System

Acknowledgement. The first author sincerely acknowledges the financial assistance received from Department of Science and Technology, New Delhi, India under Women Scientist Scheme-A vide letter number SR/WOS-A/ET-139/2011, dated 05-03-2012 and the authors are sincerely thanks the Management and Principal of Thiagarajar College of Engineering, Madurai, India to carry out this research work.

References

- [1] Voltage Stability of Power Systems: Concepts, Analytical Tools and Industry Experience, IEEE Committee, vol. IEEE/PES 93TH0358-2- PWR (1990)
- [2] Canizares, C.A. (ed.): Voltage Stability Assessment: Concepts, practices and tools. IEEE/PES Power System Stability Subcommittee Special Publication, Final Document (August 2002)
- [3] Kundur, P.: Power System Stability and Control. McGraw-Hill, New York (1994)

- [4] Srivastava, L., Singh, S.N., Sharma, J.: Estimation of loadability margin using parallel self-organizing hierarchical neural network. *Computers and Electrical Engg.* 26(2), 151–167 (2000)
- [5] Ajarapu, V., Christy, C.: The continuation power flow: a tool for steady state voltage stability analysis. *IEEE Trans. Power Systems* 7, 416–423 (1992)
- [6] Canizares, C.A., Alvarado, F.L.: Point of collapse and continuation methods for large AC/DC systems. *IEEE Trans. Power systems* 8, 1–8 (1993)
- [7] Canizares, C.A.: On bifurcation, voltage collapse and load modeling. *IEEE Trans. Power Systems* 10, 512–518 (1995)
- [8] Sode-Yome, A., Mithulanathan, N., Lee, K.Y.: A maximum Loadability Margin Method for static voltage stability in Power Systems. *IEEE Transaction on Power Systems* 21, 799–808 (2006)
- [9] Ioannis, K., Konstantinos, O.: An Analysis of Blackouts for Electric Power Transmission Systems. *Transmission on Engineering, Computing and Technology* 12, 289–292 (2006)
- [10] Kwatny, H.G., Fischl, R.F., Nwankpa, C.O.: Local Bifurcation in Power Systems: Theory, computation, and applications. *Proc. IEEE* 83(11), 1456–1483 (1995)
- [11] Bian, J., Rastgoufard, P.: Power System Voltage Stability and Security Assessment. *Electr. Power Syst. Res.* 30(3), 197–200 (1994)
- [12] Gubina, F., Strmcnik, B.: Voltage Collapse Proximity Index determination using Voltage Phasors Approach. *IEEE Transaction on Power System* 10(2), 778–794 (1995)
- [13] Moghavvemi, M., Omar, F.M.: Technique for Contingency Monitoring and Voltage Collapse Prediction. *IEEE Proceeding on Generation, Transmission and Distribution* 145(6), 634–640 (1998)
- [14] Mohamed, A., Jasmon, G.B., Yusoff, S.: A Static Voltage Collapse Indicator using Line Stability Factors. *Journal of Industrial Technology* 7(1), 73–85 (1989)
- [15] Moghavvemi, M., Faruque, O.: Real-Time Contingency Evaluation and Ranking Technique. *IEEE Proceeding on Generation, Transmission and Distribution* 145(5) (September 1998)
- [16] Chebbo, A.M., Irving, M.R., Sterling, M.J.H.: Voltage Collapse Proximity Indicator: behaviour and implications. *IEEE Proc.-C* 139(3) (May 1992)
- [17] Kessel, P., Glavitsch, H.: Estimating the Voltage Stability of a Power System. *IEEE Transactions on Power Delivery*, vol.PWRD-1(3) (July 1986)
- [18] Sinha, A.K., Hazarika, D.: A Comparative study of Voltage Stability Indices in a Power System. *Electrical Power and Energy System* 22, 589–596 (2000)
- [19] Suganyadevi, M.V., Babulal, C.K.: Estimating of Loadability Margin of a Power System by comparing Voltage Stability Indices. In: *Proceeding of IEEE International Conference on Control, Automation, Communication And Energy Conservation*, June 4-6 (2009)
- [20] Suganyadevi, M.V., Babulal, C.K.: Prediction of Loadability Margin of a Power System using Support Vector Machine. In: *Proceeding of IEEE International Conference on Energy Efficient Technologies for Sustainability (ICEETS 2013)*, April 10-12 (2013)
- [21] Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: A new learning scheme of feedforward neural networks. In: *Proc. IJCNN*, Budapest, Hungary, July 25-29, vol. 2, pp. 985–990 (2004)
- [22] Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: Theory and applications. *Neurocomputing* 70(1-3), 489–501 (2006)

- [23] Huang, G.-B., Chen, L., Siew, C.-K.: Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Netw.* 17(4), 879–892 (2006)
- [24] Huang, G.-B., Chen, L.: Convex incremental extreme learning machine. *Neurocomputing* 70(16-18), 3056–3062 (2007)
- [25] Huang, G.-B., Chen, L.: Enhanced random search based incremental extreme learning machine. *Neurocomputing* 71(16-18), 3460–3468 (2008)
- [26] Smola, A.J., Scholkopf, B.: On a Kernel-Based Method for Pattern Recognition, Regression, Approximation And Operator Inversion. *Algorithmica* 22, 211–231 (1998)
- [27] Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 1–17 (2010)
- [28] Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C., Lendasse, A.: OP-ELM: Optimally pruned extreme learning machine. *IEEE Trans. Neural Netw.* 21(1), 158–162 (2010)
- [29] Tang, X., Han, M.: Partial Lanczos extreme learning machine for single-output regression problems. *Neurocomputing* 72(13-15), 3066–3076 (2009)
- [30] Liu, Q., He, Q., Shi, Z.-Z.: Extreme support vector machine classifier. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) *PAKDD 2008. LNCS (LNAI)*, vol. 5012, pp. 222–233. Springer, Heidelberg (2008)
- [31] Frénay, B., Verleysen, M.: Using SVMs with randomised feature spaces: An extreme learning approach. In: *Proc. 18th ESANN, Bruges, Belgium, April 28-30*, pp. 315–320 (2010)
- [32] Power System Test Archive-UWEE (University of Washington), <http://www.ee.washington.edu/research/pstca>
- [33] PSAT, Power System Analysis Toolbox Version 2.1.1, <http://www.power.uwaterloo.ca/fmilano/downloads.html>

A Peer-to-Peer Particle Swarm Optimizer for Multi-objective Functions

Hrishikesh Dewan^{1,2}, Raksha B. Nayak², and V. Susheela Devi¹

¹ Department of Computer Science & Automation
Indian Institute of Science, Bangalore
`hrishikesh.dewan@csa.iisc.ernet.in`
`susheela@csa.iisc.ernet.in`

² Knowledge & Innovation
Siemens Corporate Technology & Development Center, Bangalore,
`raksha.nayak@siemens.com`

Abstract. Particle Swarm Optimization (PSO) is a well-known technique that has been used for a wide range of optimization problems. The method is inherently parallel, wherein a group of particles wander in the solution space; communicate with one another to find the best solution. Though parallel, this method has not been much experimented in peer-to-peer computing frameworks. A peer-to-peer network brings a new set of challenges but has a number of distinct properties; for example they are prone to various types of failure but can harness the unused computing cycle of a set of systems. In this paper, we illustrate such a framework, wherein the PSO method is being implemented on top of a custom peer-to-peer network. Our framework includes novel algorithms that effectively skip overwork, finds Pareto optimal solutions that are diversified and includes both load balance and fault tolerance techniques. We demonstrate the use of this new distributed optimization framework using some well-known multi-objective benchmark functions and explain its effectiveness when compared to other systems of such types.

1 Introduction

Particle Swarm Optimization (PSO) [1] is a well-known technique that has been used for a wide range of optimization problems. PSO is being modeled on the self-adaptive behavior of flocks of birds and schools of fish to identify optimal locations. The method is inherently parallel wherein a group of particles wander in the solution space; communicate with one another to find the most optimal solution. Because of being parallel and its ability to avoid local minima, PSO is being widely used to solve a large number of optimization problems. Initially, each of the particles starts with its own position and updates its neighbors of the best solution found so far. The neighbors on seeing relatively best updates, modify their position and velocity in the solution space and move towards the position of the best particle seen so far. As evident from equation 1, a position of a particle is determined collectively by its own best position and the overall global

best. To aid in propagation of each particles best position seen so far, the particles in the solution space connect among themselves. The network connectivity so created can either be of mesh topology, random, ring etc. It is important to note that in the most basic formulation of PSO method, there are three essential steps. The first step is the spread of the candidate solutions across the entire search space, the second is the step wise evaluation of the best solution found so far and the third is the modification of the velocity of each particle towards the global best solution in the previous iteration. The algorithm is parallel in step 1 and step 3 but requires a distinct synchronization point at the second step. Due to this simplicity in design, a lot of parallel approaches exploiting the steps 1 and 2 have been designed. Most of the designs [2], [3] and [4] are however based on master-slave architectures where the master solicits replies from each of the slaves for its best update, assumes the responsibility of evaluating the global best in each iteration and forwards the best result to all of the slaves. While these type of designs facilitate the use of a large number of computing machines to participate in the optimization problem, it is less efficient. The central master component is usually the point of failure and uneven load distribution may lead some slave swarms to remain idle for a large period during certain iterations. A few asynchronous update-based solutions [5] have been proposed but they are still based on master slave architecture. On the other hand, pure distributed systems such as peer-to-peer distributed systems do not have problems of this sort. There is no central authority to administer other nodes and each node is independent and acts both as a server and a client at the same time. We propose in this paper, a complete peer-to-peer particle swarm optimizer with a number of novel techniques that further reduces the convergence time for reaching the global minimum or maximum. Our distributed system is partly based on [6], and includes modifications to support multiple objective functions that have a large number of variables. The peer-to-peer system that is being exhibited in this paper also handles fault-tolerance and uniformly distributes loads across a set of machines. We evaluated our system using a simulated test bed of several dozens of machines and used the test functions as defined in [7], [8] [9] and [10] for evaluating the system.

The rest of the paper is organized as follows. Section 2 describes the related work and also shows how the work defined in the paper is different from other contemporary papers. Section 3 introduces the algorithms and the network model of our proposed system. Section 4 details the simulated environment and the experiments that were performed. It also includes the results of the experiment. Finally in Section 5, we conclude the paper alongside a note on the future extensions of the same.

2 Related Work

The closest match related to our work is defined in [11] . In [11], the authors have solved a peer-to-peer particle swarm optimizer that finds the Pareto optimal front of a set of multi-objective functions. However, it is not clear which

method is being used to find the set of solutions. In a true multi-objective function optimization, a set of dominated and non-dominated solutions are identified. In [11] no such procedures are explicitly defined. Moreover, neither is there any mention on how fault-tolerance is being achieved nor on how the load balancing of the swarm agents are carried out although it is being clearly laid out that fault-tolerance and load balancing algorithms are one of the prime problems that need to be tackled in any peer-peer application. Also, the underlying network layer as used in [11] is based on FreePastry [12], which is a prefix tree based distributed hash table implementation. Our peer-to-peer network uses an overlay network which is largely inspired from Chord [6], but we have modified the basic routing strategy to include nodes which are logically nearer in the objective space in addition to the lexicographic neighbor nodes as in Chord. Due to this, the propagation of the individual non-dominated collection of nodes among the nearby nodes is faster. In [13], a distributed particle swarm optimization is used to create the most efficient Bayesian Network directed acyclic graph(DAG). However, the allocation of work across the distributed agents follows master slave architecture and hence they are almost the same as noted in [2], [3] and [4]. In [5], a distributed PSO based approach is used to solve search problems by small robots. The technique as mentioned in [5] uses broadcasts to notify the neighboring nodes of the gbest. The search function is essentially a single objective function and there is no mention of failure or load balancing of search space. In fact in [5], it is clearly mentioned that the robots are miniature and hence should have minimum load in terms of processing and messaging. There are however, related works in the areas of distributed optimization techniques such as [14] and [15] but they are mainly focused towards other evolutionary algorithms. We skip them as they are of no direct relation to us.

3 Architecture and Algorithm

Our work can broadly be divided into four different sections. The first section relates to the design and algorithms for maintaining the network. We use an overlay network for application level messaging between the peer nodes with a precise semantics which is related to the optimization function. The second section relates to the protocols for joining and leaving the network. The third section relates to the protocols that ensure information propagation across the network and the fourth section includes the algorithms that help in maintaining a uniform load balance across the nodes and ensures fault tolerance. In each of these sub-sections, we define the different protocols/algorithms used. However, before we start describing the different algorithms and approaches, we describe first the modification of the basic algorithm to support multi-objective functions.

3.1 Modifications to Support True Multi-Objective Function Optimization

A multi-dimensional objective function can be solved in two different ways. First, convert the multiple objective functions into a single objective function and

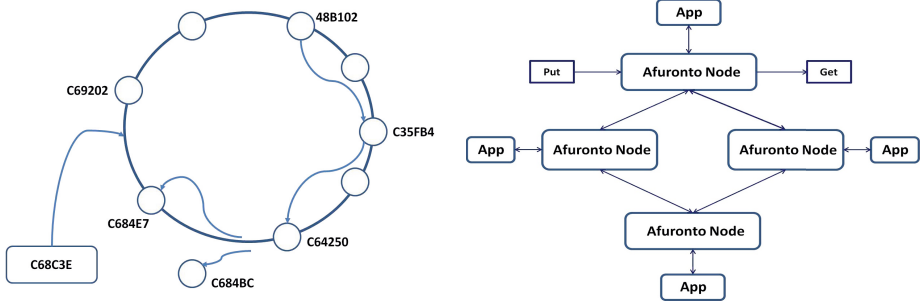
second, solve the objective functions concurrently and collectively. Clearly, using the first approach only a single optimal solution can be found and the single solution may not be the best solution from the purview of all the objective functions. Essentially it is equivalent to solving a single objective function. We have earlier defined in [16] ways of finding optimal solutions to a single objective optimization function and hence we ignore it in this paper. In this paper, we concentrate on finding solutions when multiple objective functions are collectively used to find optimal solutions. Formally, a multi-objective function can be defined as follows [9].

$$\begin{aligned}
 & \text{Minimize/Maximize } f_i(x), \text{ where } x = (x_1, x_2, x_3 \cdots x_M) \\
 & \qquad g_j(x) \leq 0, \qquad \qquad \qquad j = 1, 2, 3 \cdots J \\
 & \qquad h_k(x) = 0, \qquad \qquad \qquad k = 1, 2, 3, \cdots K \\
 & \qquad x_i = [low_i, high_i]
 \end{aligned} \tag{1}$$

We have N objective functions and M variables (noted as x). Each variable has a lower and upper bound. A number of equality and inequality constraints are also defined that restricts the variable space to a certain decision boundary. The decision boundary is therefore a multi-dimensional one and satisfies all the constraints. Since, there are multiple objective functions (of cardinality N) and variable space (of cardinality M), the decision space is mapped to an objective space. The goal in this multi-objective optimization is to find optimal solutions in this objective space. Also, since there are multiple solutions that exist, a multi-objective solution provides a number of solutions instead of just one. This set of solutions that define all the optimal solutions given a certain set of constraints and bounds constitute what is known as Pareto optimal solutions. Although finding solutions for such a set of objective functions seems straightforward, the traditional PSO algorithm does not lend itself to it. There are subtle problems and the main issue is that a PSO is inherently a single function optimizer. Therefore certain modifications are needed to handle problem statements of this kind. We illustrate the problems and consequent modification of the traditional PSO next. Unlike a single objective PSO, a multi-objective function has no single solution and hence there is no such gbest for a particular particle. Instead, gbest is the set of non-dominated solutions that are produced due to execution of the swarm. Since the number of solutions in the non-dominated set is quite large, we need efficient mechanisms to select a single solution from this set. Further, as per our goal, the Pareto optimal front that we seek to discover should be diversified in the solution space. To accommodate both of these facts, we divide the solution space in equal partitions and we tend to select the best particle from the space which is less crowded. This allows us to guide the particle towards the region which is not being explored enough and thereby find newer solutions in the process if at all there is any. Further, to reduce the overhead of storage of the non-dominated solutions, we periodically remove elements from the hyperspace. Such removals are however random.

3.2 Network Model

We use an overlay network of nodes for communication of systems as well as application level messages. The overlay network is a circular ring and is formed by applying consistent hashing functions. When a node joins the network, it is provided with a unique 160 bit identifier. The identifier is derived by concatenating time and MAC address of the node as an input to SHA-1 hash function. Based on the output, the ring is organized lexicographically. Figure 1 is an illustration of the same. For routing messages from one node to the other, each node maintains a routing table. The routing table entries are of three different types. Lexicographic neighbors (bi-directional) in both the clockwise and anti clock wise directions , unidirectional random links (based on the number of 1s in the identifier) and objective space neighbors. Objective space neighbors are the areas in function space which are adjacent to the node that is currently being computed. For 2D space, the number of neighbors is 4 and in general for l-dimensional space there are $2L$ neighbors. The reason for maintaining the objective space neighbors is to refer to it during the strategic re-work negotiation algorithm defined in the subsequent sections. When a node needs to send a message to another node, the routing table for the node is consulted to find the nearest node. Nearest node distance is the lexicographic distance. The algorithm then forwards the message to the nearest node and the same procedure is repeated till either the destination node is reached or there is no node of that identifier. In case of the former, the message is consumed by the destination node whereas in case of the latter, the message is either consumed by the most preceding node or an error is returned to the user. Due to non-uniformity of node links, the reverse path can be different from the source node. It can be proved that on an average the maximum number of hops required to transfer a message is not more than $\log K$, where K is the number of nodes in the network. Also, due to non-uniformity of processing capability, each node may compute widely disparate function spaces. As such, therefore, the objective neighbor space changes from time to time and is not bound to be static during the entire life cycle of a node. Our network differs from other DHT-based networks such as Chord [6], Pastry [12] etc. by including extra neighbor node information related to our objective function. This modification is done to decrease the message overload and also for fast convergence. When a node completes its job of finding the pareto front, it does not seek to enter into some other nodes dimensional space. Instead, the completed or the lightly loaded node simply skips the region of space and moves to the space which is slowly progressing or not yet processed. Thus, unlike the traditional PSO algorithms where there is a large amount of duplication of work, this design ensures very little duplication of work. Hence, compared to other algorithms, the relative time to convergence is faster. However, due to large number of dimensions, the effective neighbors is roughly around $2N$. This is a large number even for moderately size dimensions. To reduce the effect of introducing a large number of such neighbor nodes, we include a tree base directory structure . Every node maintains a single indirection to its parent and each parent includes the list of siblings. The number of siblings is not more than the

Table 1. Figure 1-2 An overlay network of nodes. and Peer-Peer PSO Architecture

number of dimensions that is being used to divide the hyper-plane. Thus, the maximum number of neighbors that a node has to store is dependent upon the problem statement and also on the factors of division. For example, for three dimensional vectors, we may first divide the function space in one of its axis. The second level includes the divisions of the second and so on. This increases the time to propagation but it prohibits the overcrowding of the routing table entries of the node.

3.3 Node Management

In a peer-to-peer network all nodes are symmetric and equal. Whenever a node has to participate in the function optimization computation, the first operation is the join operation. Upon completion of join, a node acquires a membership to the network and participates in all its activities, which includes function optimization, message passing, load balancing and fault-tolerance. Since each peer-node is responsible for atleast a certain chunk of work, there must be well-defined node leaving protocols. In this section, we define the node join and leave protocols. It is important to note that in an ad-hoc network with no central administrative unit, nodes can join and leave at any moment. Such abrupt changes in the network topology could be due to malfunctioning of hardware, software or the network. Hence, the join and the leave protocol must handle these extreme but major use cases as well.

As noted in section A, whenever a node joins, it first creates a new pseudo-random ID of 160 bits. After the ID is created, it then searches for an existing node in the network to communicate with. The list of existing nodes must be supplied to the new node before it starts its operation and is, by and large, a manual process. Once, the node is connected to any of the existing nodes, the first message it composes is to find duplication of its own calculated node ID. Such a message is routed to the network using the newly connected existing node. If there are no such nodes, then the node can safely join the network. On the other hand, if such a node exists, the new node creates a new ID and recursively follows the same process. Joining the network involves creating the routing table entries. After the routing table entries are successfully created, the node connects

to any node in the network for work units. Since a node already existent in the network is completely participating in the optimization process, the new node requests work from the already existent nodes. In our present protocol, we allow a new node to acquire work from a number of existing nodes and selects the work that seems appropriate. Upon selection of the work unit, the neighbors nodes are identified and the objective neighbor node work is filled. At this stage the routing table entries are complete and also the necessary code and work unit is with the new node. The new node starts executing its own sequential PSO from this point. Note that this type of work selection is completely deterministic and may take some time if the already existing nodes do not have work to allocate to a new node. The other joining protocol that we have investigated is to randomly select a portion of work from it's search space. Once the new node acquires the necessary code for the functions, its solution space and work division technique, it randomly selects a work unit from the entire available work units. Since the routing table entries maintain neighbor node information based on the decision space neighbors, a message is sent from the new node to the node which is responsible for the work chunk. If the work chunk is available, then it is allocated to the new node or the node recursively continues the random work unit selection process again until it receives the desired work load. It is important to note that if the number of nodes are larger than the available work units, then neighbor nodes can further partition its work space and allocate a few to the newly joined node.

The node leaving protocol is opposite of what is defined in the node joining protocol. The leaving node broadcast leave messages to all its neighbors and waits for a few seconds before it leaves the network. Upon leaving the network, a hole is created in the overlay network and routing table entries of the neighbor nodes are modified. Dynamic node churns are described in more detail in a later section.

3.4 Information Propagation

Information Propagation is done using the routing table entries of each node. Thus each node in the network works as a router and is responsible for forwarding messages to the closest (lexicographically) destination node for transmission. In our system, there are two types of information. The first type is the information related to swarm optimization and second type is the information for maintaining the network. Since in a multi-objective PSO, there is no individual gbest but a set of non-dominated solutions, the only information that is shared across peer swarm agents is the non-dominated set. We represent a non-dominated data set using a KD Tree and hence this data structure is transferred across the set of nodes. However, unlike gbest transfer in a PSO, we need not transfer the set of non-dominated solutions to all of the nodes. As noted earlier, our objective space is divided into distinct hyper-planes and we maintain a hierarchy of neighbors. The non-dominated solution is only passed to the neighbors. The receiving node takes the responsibility of finding dominated solutions, if any, from this set. The other information that is passed in the network is related to the maintenance of

the network. Node join and leave, heartbeat messages and that of load balancing information are some of them. More details of these messages are mentioned in their respective subsections.

3.5 Load Balancing and Fault Tolerance

A peer-to-peer ad-hoc network is always a mix of diversified components: diversified in terms of hardware resources and software components available for computation. Therefore, there is no uniformity in the completion time of a solution space. Some nodes may take a long time to compute a work unit whereas some nodes may complete the same in 1/10th of the time taken by the other node. As a result, load balancing of nodes is an important requirement in such a diversified peer-to-peer network. We balance loads not instantaneously, but after repeated step intervals. A step interval is a finite number of iterations. After completion of each step, the node propagates its load to the neighbors using a broadcast. Nodes that are lagging behind comparatively are further propagated. For broadcasting, there are two specific rules. For each node we maintain a least and utmost load, which are respectively 20% and 80 % of load capacity. If the CPU utilization falls below or above this limit, the node broadcasts this information to the neighbors. Every node therefore maintains the load of its neighbors. If, however, the load is not below or above this threshold limit, there is no message sent. Hence, load information table is not as populated as the routing table entries. Once a node receives such information, it compares the load with all the entries and tries to achieve equilibrium by matching low capacity nodes with the high capacity ones. If, on the other hand, there are nodes that are still not yet matched, then the information is passed on to the nodes neighbors. The process is repeated until either there is a match or there are no nodes to match. When a node completes its allocated work unit and there are no more pending works, the computational utilization decreases by 10%. Under these circumstances, the broadcast is sent from the node to all its neighbors till it receives new chunk of data. It is not difficult to prove that lowest utilized node is always eventually matched with a highly loaded node and the maximum number of steps required to match such a node is no more than $\log P$, where P is the number of nodes in the network. Thus the network tries to balance load at the neighborhood first and if unsuccessful, propagates the information to the next level. With this, there are no central co-coordinators required and also the number of messages required for balancing the load is less. As in the case of load balancing, fault tolerance is also handled co-operatively. Each node upon joining the network maintains three fault tolerant connections to three other nodes. These nodes need not be entries in the routing table and selected by generating three random identifiers by the node. After every successful time interval, which is configurable, the node sends the best positions and work unit in allocation information to each of these nodes. Failure to receive updates by the majority, either due to its software/hardware or network partition, signals the node as dead and a new node is selected for execution of the work.

4 Experiments and Results

For the sake of experimentation and evaluation of our proposed framework, we have created a cluster of several dozen nodes. Our main focus of the experiments is to perfectly observe the convergence achieved for functions as defined in [7], [8], [17], [9] and [10]. The entire code base is written in C# and executes in both Windows and Linux environment. We show below the graphs under normal operation for each of these functions for different nodes. We also simulate the load and fault-tolerance by artificially injecting extra time for computation and by killing processes that represent nodes. The graphs for the same together with their analysis are explained as below. The results were analyzed and several metrics as mentioned in [18] are used to evaluate the result. Apart from the test functions shown here, we have also conducted an equal number of experiments for other test functions as mentioned in [8] and [17]. For lack of space, we are omitting the results of those experiments. The experiments were run in five different modes. In the first mode, we selected 4 nodes with each node having 10 particles. Subsequently the number of machines were increased but with the same number of particles. The machines used were all commodity off-the-shelf (COTS) machines but with heterogeneous hardware configuration and software. The machine with smallest capacity is a 2 GB machine with 2 cores and the highest capacity machine has 4 cores with 4 GB of RAM. The machines were allowed to fail randomly by explicit killing of the PSO process executing in its memory. The result set is forwarded to a single machine which hosts a REST based service. The communication among the machines uses HTTP and formats the updates in JSON (Java Script Object Notation). HTTP and JSON are used for its ubiquity and acceptance in wide range of network software. Table 2 shows the results of each of the functions. The graphs as shown in Table 3 shows the convergence plots. The convergence plot is currently shown only for 50 particles. For lack of space, we have omitted the convergence plots for the other particles. For obtaining the statistical properties of average, best, worst etc, each experiment was run for 25 runs.

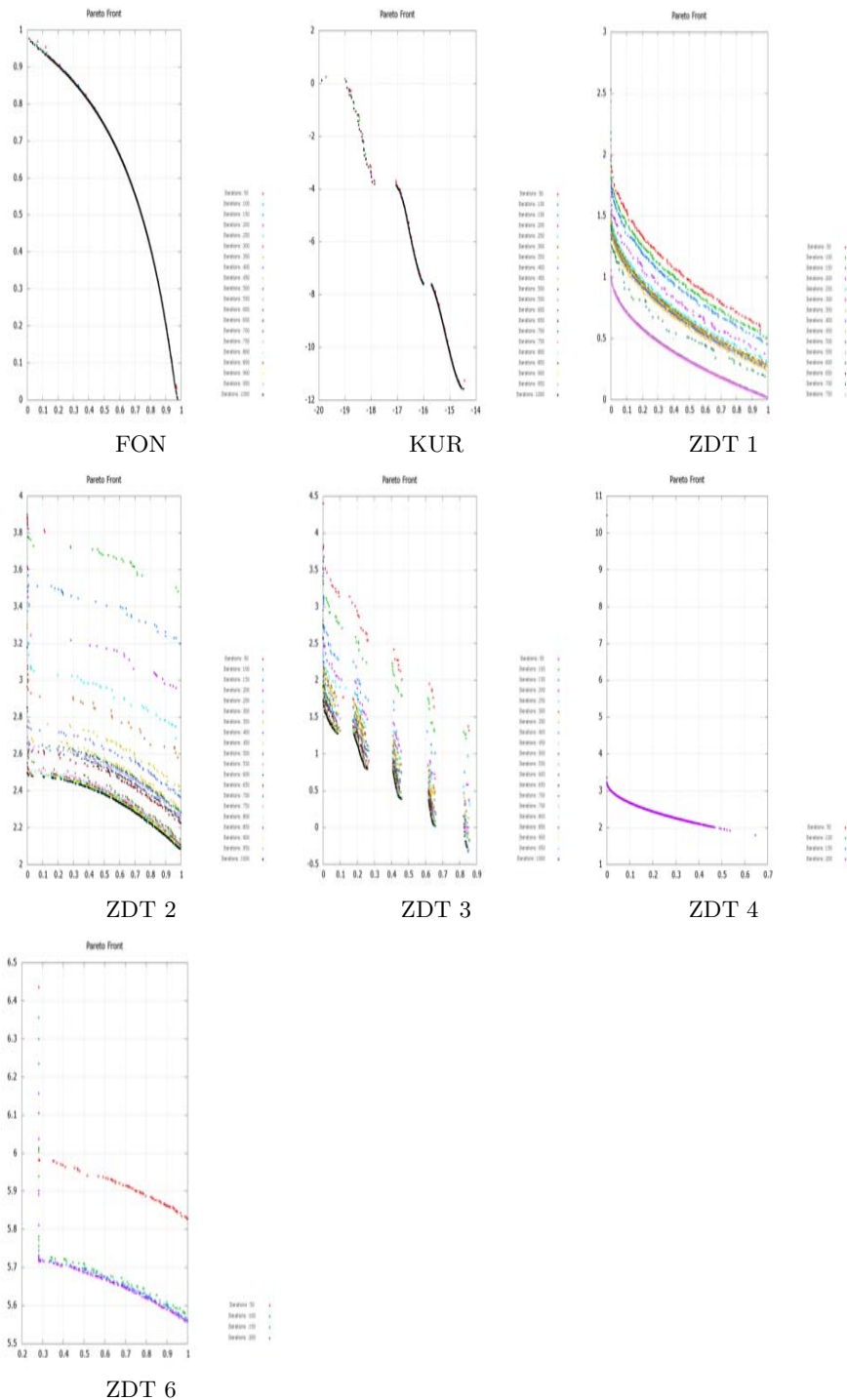
In table 3 we have shown the convergence graph for each of these particles. For each function, we have shown results with FES/particles as 25,000/50 and 1,00,000/100. Each experiment is executed for twenty five runs and the results of the twenty five runs are tabulated in Table 2. Further, each function changes its internal representation after every five iterations. Apart from these results, we have also experimented with the affect of convergence graphs for node churns and stimulated increased load. The results for the node churns are not included in this paper for lack of space.

As evident from the tables and graphs, the performance of the algorithm is acceptable and comparable with all other single process or parallel (master-slave, asynchronous) or other distributed based peer-to-peer optimizers. For the sake of experiments and due to non availability of actual test functions for large decision/objective space, we have introduced intended process sleep in between. This is done to observe the convergence of the swarm and in real practical applications such intended sleeping of threads may be entirely avoided.

Table 2. Metrics for FES: 75000, 100 particles

| KUR | | | | | | | | | |
|----------|-----------|-----------|-----------|-----------|----------|-----------|------------|-----------|-----------|
| Metric | S | ER | GD | CM | HV | Spread | Max Spread | SC | WM |
| Best | 0.0088 | 0.0000 | 10.77E-05 | 0.0047 | 32.4944 | 1.0390 | 8.8548 | 0.0135 | 0.7273 |
| Worst | 0.0549 | 0.0033 | 17.89E-05 | 0.0068 | 36.8696 | 1.5750 | 13.0153 | 0.0790 | 1.1025 |
| Average | 0.0212 | 0.0012 | 14.43E-05 | 0.0052 | 35.7881 | 1.3767 | 12.6105 | 0.0337 | 0.9637 |
| Variance | 95.15E-06 | 63.37E-08 | 43.44E-11 | 28.99E-08 | 0.8147 | 0.0276 | 0.6500 | 38.74E-05 | 0.0135 |
| Std dev | 0.0098 | 79.60E-05 | 20.84E-06 | 53.84E-05 | 0.9026 | 0.1661 | 0.8062 | 0.0197 | 0.1163 |
| ZDT 1 | | | | | | | | | |
| Best | 0.0012 | 0.0000 | 28.57E-05 | 0.0085 | 118.4877 | 0.5290 | 1.4098 | 0.9559 | 0.3704 |
| Worst | 0.1931 | 0.8541 | 0.0537 | 0.1809 | 120.5662 | 1.1173 | 4.1323 | 1.0000 | 0.7844 |
| Average | 0.0250 | 0.0544 | 0.0046 | 0.0349 | 120.3121 | 0.8448 | 2.3849 | 0.9982 | 0.5927 |
| Variance | 0.0013 | 0.0304 | 10.56E-05 | 0.0022 | 0.2728 | 0.0225 | 0.4521 | 74.74E-06 | 0.0113 |
| Std dev | 0.0366 | 0.1743 | 0.0103 | 0.0467 | 0.5223 | 0.1499 | 0.6724 | 0.0086 | 0.1061 |
| ZDT 2 | | | | | | | | | |
| Best | 0.0038 | 1.0000 | 0.0936 | 1.3880 | 81.3383 | 0.9415 | 0.1174 | 1.0000 | 0.6983 |
| Worst | 0.0760 | 1.0000 | 0.8185 | 2.6489 | 98.3533 | 1.0379 | 1.5729 | 1.0000 | 0.9423 |
| Average | 0.0311 | 1.0000 | 0.5249 | 2.0686 | 89.0527 | 0.9957 | 0.5875 | 1.0000 | 0.8545 |
| Variance | 43.59E-05 | 0.0000 | 0.0292 | 0.1117 | 18.3653 | 43.59E-05 | 0.1166 | 0.0000 | 0.0035 |
| Std dev | 0.0209 | 0.0000 | 0.1709 | 0.3343 | 4.2855 | 0.0209 | 0.3414 | 0.0000 | 0.0590 |
| ZDT 3 | | | | | | | | | |
| Best | 0.0088 | 0.5000 | 0.0124 | 0.1003 | 120.7375 | 0.8396 | 2.3519 | 1.0000 | 0.5969 |
| Worst | 0.1335 | 1.0000 | 0.0640 | 0.4795 | 126.7290 | 1.0591 | 4.3535 | 1.0000 | 0.7489 |
| Average | 0.0520 | 0.9788 | 0.0314 | 0.2837 | 123.8138 | 0.9485 | 3.2204 | 1.0000 | 0.6734 |
| Variance | 0.0010 | 0.0096 | 13.14E-05 | 0.0136 | 2.9523 | 0.0032 | 0.2962 | 0.0000 | 0.0016 |
| Std dev | 0.0323 | 0.0979 | 0.0115 | 0.1166 | 1.7182 | 0.0568 | 0.5443 | 0.0000 | 0.0403 |
| ZDT 4 | | | | | | | | | |
| Best | 53.02E-05 | 1.0000 | 0.0324 | 2.0053 | 95.1063 | 0.8456 | 1.3934 | 1.0000 | 0.6051 |
| Worst | 0.5888 | 1.0000 | 0.0742 | 2.9886 | 106.4861 | 1.4304 | 40.5834 | 1.0000 | 1.0138 |
| Average | 0.0483 | 1.0000 | 0.0451 | 2.4303 | 102.9683 | 1.0386 | 5.3675 | 1.0000 | 0.7406 |
| Variance | 0.0135 | 0.0000 | 12.14E-05 | 0.0663 | 9.8740 | 0.0256 | 60.4043 | 0.0000 | 0.0126 |
| Std dev | 0.1164 | 0.0000 | 0.0110 | 0.2574 | 3.1423 | 0.1599 | 7.7720 | 0.0000 | 0.1124 |
| ZDT 6 | | | | | | | | | |
| Best | 0.0027 | 1.0000 | 0.2552 | 2.2657 | 49.0858 | 0.9151 | 0.0303 | 1.0000 | 0.7710 |
| Worst | 0.2628 | 1.0000 | 1.4519 | 5.2836 | 90.3917 | 1.0300 | 3.1662 | 1.0000 | 1.1326 |
| Average | 0.0730 | 1.0000 | 0.6993 | 3.6662 | 70.2634 | 0.9885 | 1.1165 | 1.0000 | 0.9018 |
| Variance | 0.0046 | 0.0000 | 0.0910 | 0.6700 | 109.3330 | 70.75E-05 | 0.6970 | 0.0000 | 0.0091 |
| Std dev | 0.0677 | 0.0000 | 0.3016 | 0.8185 | 10.4562 | 0.0266 | 0.8349 | 0.0000 | 0.0952 |
| Fonseca | | | | | | | | | |
| Best | 50.28E-05 | 0.0100 | 88.52E-05 | 0.0559 | 120.0506 | 0.8505 | 1.3555 | 1.0000 | 0.5956 |
| Worst | 92.84E-05 | 0.0472 | 92.76E-05 | 0.0576 | 120.3048 | 0.9807 | 1.3880 | 1.0000 | 0.6868 |
| Average | 68.86E-05 | 0.0239 | 90.43E-05 | 0.0568 | 120.2429 | 0.9041 | 1.3780 | 1.0000 | 0.6331 |
| Variance | 14.82E-09 | 77.02E-06 | 12.87E-11 | 18.28E-08 | 0.0036 | 0.0013 | 46.61E-06 | 0.0000 | 64.03E-05 |
| Std dev | 12.18E-05 | 0.0088 | 11.34E-06 | 42.75E-05 | 0.0599 | 0.0362 | 0.0068 | 0.0000 | 0.0253 |

Table 3. Convergence Graph for FES - 75,000, 100 particles



5 Conclusion

As noted in this paper and proved by simulation, distribution of particle swarm optimizations can be effectively used to solve large scale multi-objective functions. The time to converge is little more than that of stand-alone or dedicated parallel clusters but none-the-less the use of peer-to-peer ad-hoc network clearly outweighs the benefits of clusters. This increase in time is due to the delay in message transmission of the network. However, this framework can be used in internet and such other loose structures much like the other peer-to-peer applications such as file/object storage and distributed tasks execution. Moving ahead, we foresee the following extensions to our model : creating a more efficient overlay network for fast message transmission, reducing the current delay of transmission and including security measures for protecting messages related to swarm optimization such as non-dominated solution updates and byzantine faults. We also intend to publish more elaborate results along with several parameters for these set of functions and more as mentioned in [8] and [17].

References

1. Kennedy, J.: Particle swarm optimization. In: Encyclopedia of Machine Learning, pp. 760–766. Springer (2010)
2. Chu, S.C., Roddick, J.F., Pan, J.S.: Parallel particle swarm optimization algorithm with communication strategies. submitted to IEEE Transactions on Evolutionary Computation (2003)
3. Schutte, J.F., Reinbolt, J.A., Fregly, B.J., Haftka, R.T., George, A.D.: Parallel global optimization with the particle swarm algorithm. *International Journal for Numerical Methods in Engineering* 61(13), 2296–2315 (2004)
4. Parsopoulos, K.E., Tasoulis, D.K., Vrahatis, M.N., et al.: Multiobjective optimization using parallel vector evaluated particle swarm optimization. In: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004), vol. 2, pp. 823–828 (2004)
5. Hereford, J.M.: A distributed particle swarm optimization algorithm for swarm robotic applications. In: IEEE Congress on Evolutionary Computation, CEC 2006, pp. 1678–1685. IEEE (2006)
6. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: ACM SIGCOMM Computer Communication Review, vol. 31, pp. 149–160. ACM (2001)
7. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation* 8(2), 173–195 (2000)
8. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multiobjective 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)
9. Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation* 3(1), 1–16 (1995)
10. Kursawe, F.: A variant of evolution strategies for vector optimization. In: Schwefel, H.-P., Männer, R. (eds.) *Parallel Problem Solving from Nature*. LNCS, vol. 496, pp. 193–197. Springer, Heidelberg (1991)

11. Scriven, I., Lewis, A., Mostaghim, S.: Dynamic search initialisation strategies for multi-objective optimisation in peer-to-peer networks. In: IEEE Congress on Evolutionary Computation, CEC 2009, pp. 1515–1522. IEEE (2009)
12. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Guerraoui, R. (ed.) *Middleware 2001*. LNCS, vol. 2218, pp. 329–350. Springer, Heidelberg (2001)
13. Sahin, F., Yavuz, M.Ç., Arnavut, Z., Uluçol, Ö.: Fault diagnosis for airplane engines using bayesian networks and distributed particle swarm optimization. *Parallel Computing* 33(2), 124–143 (2007)
14. Tan, K.C., Yang, Y., Goh, C.K.: A distributed cooperative coevolutionary algorithm for multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 10(5), 527–549 (2006)
15. Hiroyasu, T., Miki, M., Watanabe, S.: The new model of parallel genetic algorithm in multi-objective optimization problems-divided range multi-objective genetic algorithm. In: *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 1, pp. 333–340. IEEE (2000)
16. Dewan, H., Devi, V.S.: A peer-peer particle swarm optimizer. In: *2012 Sixth International Conference on Genetic and Evolutionary Computing (ICGEC)*, pp. 140–144. IEEE (2012)
17. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: *Proceedings of the Congress on Evolutionary Computation (CEC-2002)*, Honolulu, USA, pp. 825–830 (2002)
18. Deb, K.: Multi-objective optimization. *Multi-Objective Optimization Using Evolutionary Algorithms*, 13–46 (2001)

A Novel Improved Discrete ABC Algorithm for Manpower Scheduling Problem in Remanufacturing

Debabrota Basu¹, Shantanab Debchoudhury¹,
Kai-Zhou Gao², and Ponnuthurai Nagaratnam Suganthan²

¹Dept. of Electronics & Telecommunication Engineering,
Jadavpur University, Kolkata, India

²School of EEE, Nanyang Technological University, Singapore, 639798
basudebabrota29@gmail.com

Abstract. Remanufacturing technique is a widely used approach in modern industries. But the very first step of this technique is disassembling. This disassembling operation requires an efficient employee pool and their allocation to several steps of disassembling. In this paper, we have proposed an improved ABC algorithm that can be used to solve the manpower scheduling problem for the disassembling operation in remanufacturing industry. We test this algorithm on several instances along with some existing state-of-art algorithms. The results prove the efficiency of this algorithm to solve manpower scheduling problem in remanufacturing.

Keywords: Remanufacturing, Man-power scheduling, Artificial Bee Colony algorithm, ID-ABC.

1 Introduction

Remanufacturing is a process of product recovery used in industries [1]. In this process, used products or modules are disassembled to its basic components among which the faulty components are cleaned, repaired and replaced with new ones. After this process of disassembling and repairing these new sets of basic components are reassembled to produce the main product or module which will be in sound working condition.

In this point of time when humanity is facing the problem of global warming and the lack of renewable energy and resources remanufacturing is an important process. Remanufacturing can help us to use the resources again and again without any degradation of product quality and performance. Thus, remanufacturing is considered as the *ultimate form of recycling* and now-a-days it is an interesting topic for researchers [2]. Even remanufacturing is practically used in several countries across the globe for remanufacturing of several products like aerospace, air-conditioning units, bakery equipments, computer and telecommunication equipment, defence equipments, robots, vending machines, motor vehicles and many more. This environmental edge of remanufacturing process [3] has inspired us to deal with this problem.

But efficient disassembling of these products or modules require an efficient employee pool who will have a knowledge of these equipments and the knowledge of steps through which the product or the module can be disassembled to its basic components from which it can be reassembled to build up the actual product which can work as efficient as a new one. Here comes the problem of scheduling man force in disassembling process of remanufacturing industry. Because after one product or its part is divided into sub-components then disassembling of those sub-components become independent of each other. Besides this, every employee cannot do every operation of the disassembling. Rather there exists a certain set of employees with certain skill sets for each of the operations involved in disassembling. Thus, the parallelism and presence of constraints make this problem a challenging combinatorial optimization problem which can hardly be solved by hands for large or real time cases.

There are several heuristics and meta-heuristics which have already been developed to produce efficient solutions for several combinatorial problems like job shop scheduling problems, man power scheduling problem[4]-[6] etc. Shifting Bottleneck Procedure [7] has proved to be the most successful heuristic for scheduling problems. But the shortcoming of these algorithms is they are too much problem specific. But due to robustness and exceptional performance of several bio-inspired algorithms, now-a-days new meta-heuristic approaches are invented and used for solving these scheduling problems. Goldberg [8] first used Genetic Algorithm to solve the scheduling problems. Ho *et al* [9] used modified Tabu Search technique to solve man power scheduling problem for airline catering. But there is hardly any significant work in solving the manpower scheduling for disassembling industry where the problem follows a parallel tree pattern rather than two ended graphical pattern followed by other scheduling problems.

Here, we have proposed a novel meta-heuristic approach, named as Improved Discrete ABC (ID-ABC) based on ABC algorithm to solve this problem. Artificial Bee Colony (ABC) algorithm [10] is an efficient global optimization tool which was first developed by Karaboga *et al*. Though it was first proposed as a continuous optimizer, later researchers have developed several modifications of it and even developed discrete versions of it to solve several combinatorial optimization problems like job shop scheduling problem [11]-[12], travelling salesman problem [13] etc.

Rest of the article is presented as follows: in the next section, we have described the mathematical formulation of the man power scheduling problem for disassembling process using integer programming. After that in section 3, the basics of ABC algorithms is described in brief and in section 4, the improved discrete ABC algorithm (ID-ABC) is proposed with its pseudo code. In the very next section, this algorithm is tested on several problem instances and the result is compared with several state-of-art algorithms, which proves the efficiency of this algorithm to solve the scheduling problem. In section 6 we have stated the conclusion of our paper where we have explored the efficiency of our algorithm and the possible future works that can be done using this model and algorithm.

2 Mathematical Modeling of Scheduling Problem

Let us consider the following tree structure for the arrangement of operations in the disassembling process where a single product or module has to be divided in n basic components through a set of operations \mathbb{O} . Here, we may consider each of the operations as the nodes of the tree where the main product will be the root of the tree and the final components will be the leaves. Now, there is a set of m employees \mathbb{E} working on this process where every node i has its own set of eligible employees E_i such that, $E_i \subseteq \mathbb{E}$ and $\bigcup_{i \in \mathbb{O}} E_i = \mathbb{E}$. The time required for the j^{th} employee to perform the i^{th} operation is given by t_{ij} , where $j \in E_i$. To clarify the problem statement let us consider the following example: there are 4 employees and a product has to be disassembled into 4 basic components, where the disassembling process will follow the tree as shown in fig 1. Where 5 leaves represent 5 basic components that has to be derived from the original one and the root node or operation 0 represents the basic module.

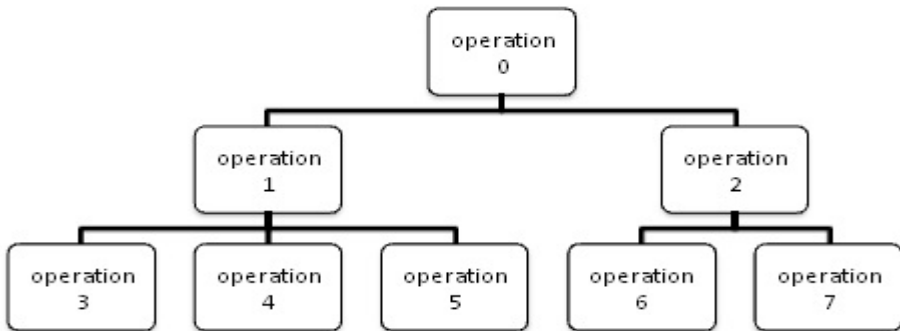


Fig. 1. The disassembly operation tree for 5 components, 4 employee problem

Now, the tree shown above contains 7 nodes representing 7 disassembled modules where each of them have their own eligible employee list and processing time list for each of them. As shown in table 1 below:

Table 1. List of eligible employees and corresponding processing times for each operation

| <i>Operations</i> | <i>Employee(Processing_time)</i> | | |
|-------------------|----------------------------------|-------|-------|
| 1 | 4(10) | 3(15) | |
| 2 | 1(5) | 3(10) | |
| 3 | 1(6) | 2(9) | 4(8) |
| 4 | 3(7) | 4(10) | |
| 5 | 2(5) | 3(6) | |
| 6 | 2(4) | 4(7) | |
| 7 | 1(11) | 3(8) | 4(10) |

Before presenting the problem statement let us state the basic problem statements and assumptions of this problem:

1. Each employee is ready to perform the operation at the time $t=0$ and there is no time they need to take between the two successive operations.
2. Two operations belonging to two different subtrees are independent of each other but the operations belonging to the same subtree are not independent because it cannot be fulfilled until its parent unit (node) is disassembled.
3. No interruption of an operation is allowed-each must be scheduled into a single contiguous time interval.
4. The processing times, the employee list and the operation division tree are known previously, as shown in fig. 1. and table 1.
5. There is a certain non-zero waiting time for the employees working on the nodes after the level 1 of the tree. We can represent the waiting time for j^{th} employee to perform the i^{th} operation is given by w_{ij} , where $j \in E_i$ and

$$w_{ij} = \max \left\{ \sum_{i \in C} p_{ij}, \sum_{i \in Q} p_{ij} \right\} \quad \text{where, } C = \text{set of all nodes from } i^{th} \text{ operation to the root node} \tag{1}$$

$Q = \text{set of all nodes from where } j^{th} \text{ employee is assigned before } i^{th} \text{ operation}$

Now, if S is a possible and valid permutation of the operations and corresponding employee list, then the makespan of the disassembling process can be described as:

$$C(i) = p_{ij}, \quad \text{if the operation } i \text{ is in the level 1} \tag{2}$$

$$C(i) = p_{ij} + w_{ij}, \quad \text{if the operation } i \text{ is not in the level 1} \tag{3}$$

$$C(S) = \max \{C(i)\}, \quad \text{if the operation } i \text{ is a leaf i.e., basic component} \tag{4}$$

The objective of our problem is to find such a permutation of operations and employees S , such that the makespan $C(S)$ will be minimum.

3 Artificial Bee Colony Algorithm (ABC)

Artificial bee colony algorithm (ABC) is one of the most popular and efficient swarm algorithm developed in recent times [10]. In this algorithm three bee phases are repeated iteratively until the termination condition is reached. The employed bees are responsible for exploiting the food sources, and they share food sources information with onlooker bees which are waiting in the hive to make the decision to choose food sources. This phase acts as a positive feedback mechanism; whereas scout bees carry out a random search near the hive for new food sources, which activates the negative

feedback mechanism. Each food sources are represented by n-dimensional real-valued vectors which denote solutions to the problem under consideration, whereas the nectar amount of the food resource is evaluated by the fitness value.

Now, movement of onlookers is controlled by probability of selecting the source, which is given by the following equation,

$$p_k = \frac{f_k}{\sum_{k=1}^{SN} f_k} \quad (5)$$

where, p_k : probability of selecting the k^{th} employed bee, SN: no. of employed bees, and, f_k : the fitness value of the position where k^{th} employed bee is placed.

Again, depending on the nectar densities as shown above the new positions are calculated as follows,

$$x_{ij}(t+1) = x_{ij}(t) + \phi_{ij} * (x_{ij}(t) - x_{kj}(t)) \quad (6)$$

x_i : Position of the onlooker bee, t: iteration number, x_k :randomly chosen employed bee, j : dimension of the solution, ϕ_{ij} : series of random variables in the range [-1,1].

This step is known as greedy selection strategy.

But, to activate the negative feedback mechanism and to keep the exploration property of ABC intact movement of scout bees is conducted according to,

$$x_{ij} = \min_j + \text{rand}(0,1) * (\max_j - \min_j) \quad (7)$$

4 Improved Discrete ABC (ID-ABC)

ABC algorithm is actually not formulated for discrete domains. It works in the continuous domain with its unique positive and negative feedback mechanism which makes it a strong optimization tool. But here our problem is mainly a combinatorial optimization problem. To adapt ABC algorithm in this discrete domain and to work effectively on this problem we have used some novel techniques with the state of art ABC algorithm which evolves this Improved Discrete ABC (IDABC) algorithm. The key processes of IDABC are discretization of ABC algorithm for the problem described in section (2), destruction and construction based iterative greedy (IG) heuristic [14] and the RIS local search algorithm with priority rules, which are discussed elaborately in the next subsections.

4.1 Initialization

In this algorithm, we have considered each food source S_k as a job vector \vec{J}_k consisting of a sequence of jobs which has correspondence with another employee vector

\vec{W}_k , where corresponding employees are selected for the jobs are sequenced. We can express it mathematically as follows:

$$S_k = (\vec{J}_k, \vec{W}_k) \quad \text{where } k = 1, 2, \dots, NP \quad (7)$$

where, $\vec{J}_k = (j_1, j_2, \dots, j_n)$ where, $j_i \in \mathbf{O}$

and, $\vec{W}_k = (e_1, e_2, \dots, e_n)$ where, $e_i \in E_{j_i}$

The solution space which includes NP such food sources is generated randomly. Here, each of the food sources is also randomly assigned with a destruction size d_k . This destructor size is allocated on the food sources as they will generate their neighbors depending on the destruction and construction process of iteratively greedy (IG) heuristic which is discussed in the section (4.5).

4.2 Employed Bee Phase

The employed bees generate food sources in the neighbourhood of their current positions. It is basically the search phase of ABC algorithm where the swarm explores the search space. For neighbour generation, we have taken three different methods: insert, swap and construction destruction with size d_k . In case of every employed bee of the population any one of the above perturbation methods are adopted in a random manner. It enriches the neighbourhood structure and diversifies the population. The swap operator randomly selects two positions on the corresponding food sources and interchanges their corresponding positions. The insert operator takes any random position of the food source and inserts it somewhere else in the food source randomly. The destruction and construction method is as shown below,

Pseudocode 1. Destruction and Construction Process

```

1 //Destruction Process//
  for iter = 1 to  $d_k$ 
     $S_{Dk}$  = remove one job and the corresponding employee
              from the sequence  $S_k$  and insert it into the
              sequence  $S_{Rk}$ 
  end
2 //Construction Process//
  for iter = 1 to  $d_k$ 
     $S$  = best permutation obtained by inserting  $i^{\text{th}}$  job
          and the corresponding employee pair from the
          sequence  $S_{Rk}$  and insert it into the sequence  $S_D$ 
  end

```

After generating a pool of such food sources, RIS local search along with the priority rules of scheduling is applied to further improve the solution quality.

4.3 Onlooker Bee Phase

In this phase, the greedy selection process is applied on the food sources to select out the best set of food sources. Generally, systems like roulette wheel are used here to select the best set of food sources. The onlooker bees utilize the same method as used by the employed bee to produce a new neighbouring solution S_{new} in the neighbour of the selected food sources. Then among the final pool of food sources the selection is done on the basis of tournament selection. Here, 2 of the food sources will be chosen randomly and the source with minimum makespan will be retained for the next step. Again, RIS search with priority is used on them to give legal and good quality solutions.

4.4 Scout Bee Phase

In the basic ABC algorithm, a scout bee produces a food source randomly in the pre-defined search space. This will decrease the search efficacy, since the best food source in the population often carries better information than others during the search process, and the search space around it could be the most promising region. Therefore, in the DABC algorithm, a tournament selection with the size of two is again used to discard a solution in such a way that two random food sources are picked up from the population and the worst one is selected. Then the scout generates a food source by performing a DC with the best destruction size d_{best} to the best solution in the population and replaced with the food source determined by tournament selection.

4.5 IG Algorithm

In the IG algorithm used here, the destruction phase is concerned with removing randomly d number of jobs and corresponding employee components from a previously constructed solution S to generate two partial solutions S_R and S_D , whereas construction phase is related to the reconstruction of a complete solution S_{new} , by using a greedy constructive heuristic. Here, we have employed NEH heuristic as described in [15]. An acceptance criterion is then used to decide whether or not the reconstructed solution will replace the incumbent solution.

Pseudocode 2. Iterative Greedy algorithm

```

1  $S_0$  = initially generated solution
2  $S$  = Local Search ( $S_0$ )
3 While (termination condition is not met)
     $S_D$  = Destruction ( $S$ )
     $S_{new}^*$  = Construction ( $S_D, S_R$ )
     $S_{new}$  = Local Search ( $S_{new}^*$ )
     $S$  = Acceptance_criterion ( $S, S_{new}$ )
end while
```

4.6 Reference Insertion Search With Priority Rules for Constraint Handling

The main purpose of the local search is to generate a better solution or food source from the neighbourhood of a given solution or food source. In this paper we have adopted a reference insert neighbourhood based local search, which has been regarded as superior to the swap or exchange neighbourhood [16].

Pseudocode 3. RIS Local search algorithm with priority rule

- 1 Set $i = 1$
 - 2 Set $\vec{J} = (j_1, j_2, \dots, j_n)$, $\vec{W} = (e_1, e_2, \dots, e_n)$ and $\mathbf{S} = (\vec{J}, \vec{W})$
 - 3 Extract a certain job j_i randomly without repetition from \vec{J} and remove it from permutation. In a similar manner the corresponding employee will be extracted from the employee vector \vec{W} .
 - 4 For $k = 1$ to n
 - a Re-insert j_i in another different position of the permutation and adjust permutation accordingly by not changing the relative positions of the other jobs to get \mathbf{S}^*
 - b If \mathbf{S}^* is better than \mathbf{S} , then
 let $\mathbf{S} = \mathbf{S}^*$
 - 5 $i = i + 1$.
 - 6 If $i \leq n$, then go to step 2
 else stop.
-

Here, we have to decide whether \mathbf{S}^* is better than \mathbf{S} or not. For that we have used the superiority of feasible (SF) approach [17] for constrained optimization, where the priority of a certain solution, here permutation or food source, is decided based on lexicographic ordering and constraint violation and objective function value are distinguished. The aim of this approach is to optimize constraint optimization problems by a lexicographic order where constraint violation gets higher priority over objective function value.

According to superiority of feasible (SF) approach the thumb-rules to decide the superiority of a certain solution \mathbf{S} over \mathbf{S}^* can be given as follows:

- 1) \mathbf{S}^* is feasible and \mathbf{S} is not.
- 2) \mathbf{S}^* and \mathbf{S} are both feasible and \mathbf{S}^* has a smaller objective function value than \mathbf{S} .

Thus, finally we can summarize the total MoDABC algorithm as follows:

Pseudocode 4. IDABC algorithm

```

1 Set parameters
2 //Initialization//
   Establish initial populations randomly with NP food
   sources where each food source S contains two n dimen-
   sional vectors,  $\mathbf{J}$  presenting a sequence of jobs and  $\mathbf{E}$ 
   a sequence of employee for the corresponding jobs.
3 Assign a destruction size  $d_i$  to each food source in the
   population which is assigned randomly in the range
   [1, n-1].
4 Evaluate population and find  $S_{best}$  and  $d_{best}$ .
5 //Employed Bee Phase//
   Repeat the following for each employed bee  $S_i$ 
   a Generate a new food source by  $S_{new} = DC_i(S_i, d_i)$ 
   b Apply local search algorithm to  $S_{new}$ 
   c if  $f(S_{new}) < f(S_i)$ ,  $S_i = S_{new}$ 
       else  $d_i = \text{rand}() \% (n-1)$ 
   d if  $f(S_{new}) < f(S_{best})$ ,  $S_{best} = S_{new}$  and  $d_{best} = d_i$ 
6 //Onlooker Bee Phase//
   Repeat the following for each onlooker bee  $S_k$ 

   a Select a food source  $S_k = \text{Tournament Selection}(S_k \in \mathcal{S})$ 
   b Generate a new food source by  $S_{new} = DC_k(S_k, d_k)$ 
   c Apply RIS local search to  $S_{new}$ 
   d if  $f(S_{new}) < f(S_k)$ ,  $S_k = S_{new}$ 
       else  $d_k = \text{rand}() \% (n-1)$ 
   e if  $f(S_{new}) < f(S_{best})$ ,  $S_{best} = S_{new}$  and  $d_{best} = d_i$ 
7 //Scout Bee Phase//
   Repeat the following steps for each scout bee  $\pi_k$ 

   a Select a food source  $S_k = \text{Tournament Selection}(S_k \in \mathcal{S})$ 
   b Generate a new food source by  $S_{new} = DC_k(S_k, d_{best})$ 
   c Apply VNS local search to  $S_{new}$ 
   d if  $f(S_{new}) < f(S_k)$ ,  $S_k = S_{new}$ 
       else  $d_k = \text{rand}() \% (n-1)$ 
   e if  $f(S_{new}) < f(S_{best})$ ,  $S_{best} = S_{new}$ 
   f if the stopping criterion is not met, got to Step 5,
       else stop and return  $S_{best}$ .

```

5 Experimental Results and Discussions

We have tested our proposed algorithm on 10 different problem instances generated in accordance with real time circumstances each with different employee numbers and different final components. Four are relatively easy scheduling problems where number of operators is limited to a maximum of 6 and depth of disassembling is at most 4. Four cases require assignment of medium level of difficulty where depth of tree is at most 7 and number of operators involved is at most 10. The remaining two cases pose a difficult assignment challenge with involvement of 15 operators and a depth level of 10.

Table 2. Comparison of results of MoDABC and other algorithms for several instances

| Instance No. | Makespan obtained and algorithmic run time | | | | | |
|--------------|--|--------------|----------|--------------|----------|---------------|
| | GA | | ABC | | ID-ABC | |
| | Makespan | RunTime(sec) | Makespan | RunTime(sec) | Makespan | RunTime (sec) |
| 1 | 20 | 0.59634 | 20 | 0.54972 | 20 | 0.50070 |
| 2 | 45 | 0.63181 | 35 | 0.52531 | 35 | 0.50090 |
| 3 | 45 | 0.75019 | 36 | 0.50924 | 30 | 0.50281 |
| 4 | 41 | 0.80111 | 41 | 0.73677 | 41 | 0.68710 |
| 5 | 135 | 1.33270 | 101 | 1.17296 | 100 | 0.99364 |
| 6 | 110 | 1.23306 | 83 | 1.23467 | 83 | 1.06776 |
| 7 | 116 | 1.10739 | 85 | 0.97186 | 85 | 0.82784 |
| 8 | 114 | 1.11338 | 83 | 0.96689 | 83 | 0.92827 |
| 9 | 198 | 1.76803 | 128 | 1.57978 | 125 | 1.66963 |
| 10 | 189 | 1.89628 | 157 | 1.87293 | 157 | 1.74956 |

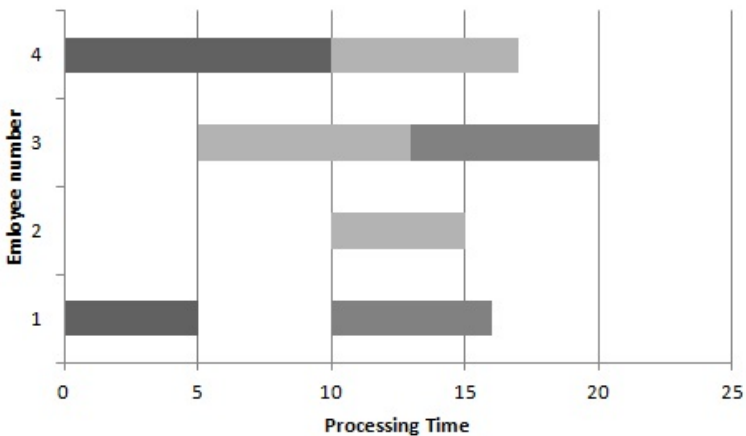


Fig. 1. Gantt chart obtained using ID-ABC for a 5components, 4 employee problem

The efficiency of ID-ABC is shown in table 2 by comparing its results with state-of-art algorithms like ABC and GA. Besides this, the gnat-chart [18] for instance number 1 with 5 employee and 4 components generated by ID-ABC is shown in the fig 2.

The parameters used in the MoDABC algorithm vary over a wide range depending on the problem dimensions. But we have generally taken the dimension of each food source as the number of operations in the problem, total number of food sources NP as the product of total number of components and the number of employee. But there is no restriction like that but it gives kind of thumb rule for the population size. Again, the number of onlooker bee is twice of NP and the size of the scout population is $0.25*NP$. The other two algorithms are also run with the same population sizes for each problem.

6 Conclusions and Future Possibilities

In this paper, we have proposed the manpower scheduling for disassembling process which is one of the foremost and basic step of remanufacturing industry. We have also presented a tree type structure and a mathematical formulation of makespan for this problem. This formulation opens a scope of application of several algorithms in this problem and using this model we can also develop many efficient heuristics and meta-heuristics in future which can solve the problem efficiently. Here, we have proposed a meta-heuristic ID-ABC which proves itself very efficient to solve this scheduling problem when compared with other state-of-art algorithms. But still there is scope of improvements in this algorithm which will make it more robust and applicable for even other discrete problems too. We can mold this algorithm with some efficient local search heuristic to make it more efficient for other combinatorial optimization problems. So, both the problem and the algorithm can be considered as future topics for research.

References

1. Remanufacturing, link: <http://en.wikipedia.org/wiki/Remanufacturing>
2. The Remanufacturing institute, link: http://reman.org/AboutReman_main.htm
3. <http://www.remanufacturing.org.uk/>
4. Lau, H.C.: On the complexity of manpower scheduling. *Computers Ops. Res.* 23(1), 93–102 (1996)
5. Shahrezaei, P.S., Moghaddam, R.T., Kazemipoor, H.: Solving a multi-objective multi-skilled manpower scheduling model by a fuzzy goal programming approach. *Applied Mathematical Modelling* 37, 5424–5443 (2013)
6. Pan, Q.K., Suganthan, P.N., Chua, T.J., Cai, T.X.: Solving manpower scheduling problem in manufacturing using mixed-integer programming with a two-stage heuristic algorithm. *Int. J. Adv. Manuf. Technol.* 46, 1229–1237 (2010)
7. Joseph, A., Egon, B., Daniel, Z.: The Shifting Bottleneck Procedure for Job-shop Scheduling. *Management Science* 34(3) (March 1988) (printed in USA)

8. Goldberg, D.E.: Genetic Algorithms in search. Addison-Wesley (1994)
9. Ho, S.C., Leung, J.M.Y.: Solving a manpower scheduling problem for airline catering using metaheuristics. *European Journal of Operational Research* 202, 903–921 (2010)
10. 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)
11. Li, J., Pan, Q., Xie, S., Wang, S.: A hybrid artificial bee colony algorithm for flexible job shop scheduling problems. *Int. J. of Computers, Communications & Control* VI(2), 286–296 (2011)
12. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review*, 1–37 (2012)
13. Li, L., Cheng, Y., Tan, L., Niu, B.: A Discrete Artificial Bee Colony Algorithm for TSP Problem. In: Huang, D.-S., Gan, Y., Premaratne, P., Han, K. (eds.) *ICIC 2011*. LNCS, vol. 6840, pp. 566–573. Springer, Heidelberg (2012)
14. Jacobs, L.W., Brusco, M.J.: A local search heuristic for large set-covering problems. *Naval Research Logistics Quarterly* 42(7), 1129–1140 (1995)
15. Nawaz, M., Ensore Jr., E.E., Ham, I.A.: Heuristic algorithm for the m-machine, n-job flow shop sequencing problem. *OMEGA* 11(1), 91–95 (1983)
16. Ruben, R., Stutzle, T.: An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *Eur. J. Oper. Res.* 187, 1143–1159 (2008)
17. Deb, K.: An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering* 186, 311–338 (2000)
18. Geraldi, J., Lechter, T.: Gantt charts revisited: A critical analysis of its roots and implications to the management of projects today. *International Journal of Managing Projects in Business* 5(4), 578–594 (2012)

Optimal Partial-Retuning of Decentralised PI Controller of Coal Gasifier Using Bat Algorithm

Rangasamy Kotteeswaran¹ and Lingappan Sivakumar²

¹Department of Instrumentation and Control Engineering,
St. Joseph's College of Engineering, Chennai India
kotteswaran@gmail.com

²Formerly General Manager(Corporate R&D), BHEL, Hyderabad, Presently Vice Principal,
Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu, India
lingappansivakumar@gmail.com

Abstract. In the recent past Metahuristic algorithms are most widely used in process industries for providing optimum outputs under certain constraints. Almost all the industrial processes are multivariable in nature with strong interactions and nonlinearities. For such processes producing optimum response is cumbersome with conventional optimization algorithms while metahuristic algorithms provide better solution. In this paper BAT algorithm, a recently developed metahuristic algorithm is used to obtain optimum response of coal gasifier which is a highly nonlinear multivariable process having strong interactions among the control loops. The existing controller along with its tuned parameters does not able to satisfy the constraints at 0% load for sinusoidal pressure disturbance otherwise this seems to be fine. The parameter of pressure loop PI controller is retuned using BAT algorithm and the performance tests are conducted. Test results shows that the retuned controller provides better response, meeting all the constraints at all load conditions.

Keywords: Bat Algorithm, Coal gasifier, Coal quality, Integrated Gasification Combined Cycle, Metahuristic algorithm, Optimization.

1 Introduction

Integrated Gasification Combined Cycle (IGCC) is an efficient method of clean power and energy generation. Here Coal reacts with air and steam, converted into coal gas (syngas or producer gas) under certain pressure and temperature. Purified Coal gas runs the gas turbine to generate power and exhaust gas from the gas turbine enters Heat Recovery Steam Generator (HRSG) to produce steam which in turn runs the steam turbine. And thus the efficiency of IGCC based power plant has increased efficiency compared to coal fired thermal power plants. Coal gasifier, an important and primary element in IGCC, which converts coal into coal gas, is a highly non-linear, multivariable process, having five controllable inputs few non-control inputs and four outputs with a high degree of cross coupling between them. The process is a four-input, four output regulatory problem for the control design (keeping limestone

at constant value). It exhibits a complex dynamic behaviour with mixed fast and slow dynamics and it is highly difficult to control. The full model of coal gasifier has 25 states and the ultimate requirement is to find the controller constants (K_p , K_i) of decentralized PI controller such that all the constraints are met for all the specified loads as given in the challenge problem [1]. Control specification includes sink pressure step and sinusoidal disturbance tests (at the three different operating points), ramp change in load from 50% to 100%, and coal quality ($\pm 18\%$) test. Input-output constraints and performance specifications are detailed [1]. A group of researchers have attempted to analyze and designed controllers and retuned the baseline controller to meet the performance objectives at all the load conditions [2-10] in the recent past. Apart from the conventional techniques, soft computing approaches such as MOGA [11] and NSGA II [12] are also used to design the controller. But still there is a scope for new metaheuristic algorithms to be used to get the optimum response.

2 Mathematical Representation of Coal Gasifier

The nonlinear model of coal gasifier can be transformed into linear model and the linearized transfer function model of the gasifier can be represented as;

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} & G_{15} \\ G_{21} & G_{22} & G_{23} & G_{24} & G_{25} \\ G_{31} & G_{32} & G_{34} & G_{34} & G_{35} \\ G_{41} & G_{42} & G_{43} & G_{44} & G_{45} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} + \begin{bmatrix} G_{d1} \\ G_{d2} \\ G_{d3} \\ G_{d4} \end{bmatrix} \times d \tag{1}$$

Where,

- G_{ij} =transfer function from i^{th} input to j^{th} output
- y_1 = fuel gas caloric value (J/kg); y_2 =bed mass (kg)
- y_3 =fuel gas pressure (N/m²); y_4 =fuel gas temperature (K)
- u_1 =char extraction flow (kg/s); u_2 =air mass flow (kg/s)
- u_3 =coal flow (kg/s); u_4 =steam mass flow (kg/s)
- u_5 =limestone mass flow (kg/s); d =sink pressure (N/m²)

Limestone flow rate is fixed at 1/10th of coal flow rate and thus the process can be reduced to 4X4 MIMO process for control purpose. For a multivariable process decentralised control schemes are usually preferred. Equation 2 shows the structure of decentralised controller used in gasifier control [1]. It employs three PI controllers and one feedforward+feedback controller for coal flow rate.

$$G_c(s) = \begin{pmatrix} 0 & \left(K_p + \frac{1}{\tau_i s} \right) & 0 & 0 \\ K_f & 0 & K_p & 0 \\ 0 & 0 & 0 & \left(K_p + \frac{1}{\tau_i s} \right) \\ \left(K_p + \frac{1}{\tau_i s} \right) & 0 & 0 & 0 \end{pmatrix} \tag{2}$$

Where, K_p =proportional gain; τ_i =Integral time and K_f =feedforward gain

The given controller structure with provided controller parameters satisfies the performance requirements at 50% and 100% operating points but fails to satisfy the constraints at 0% load for sinusoidal pressure disturbance(i.e. PGAS exceeds the limit of ± 0.1 bar). The decentralised controller may be re-tuned to meet the desired performance requirement at all load conditions and for all pressure disturbance tests. The input and outputs should be maintained under certain limits for the proper operation of gasifier. The input actuator flow limits and rate of change of limit are associated with the physical properties of the actuator, should not exceed as shown in table 1.

Table 1. Input limits

| Input variable | Max(kg s ⁻¹) | Min(kg s ⁻¹) | Rate(kg s ⁻²) |
|-------------------------|--------------------------|--------------------------|---------------------------|
| Coal inlet flow (WCOL) | 10 | 0 | 0.2 |
| Air inlet flow (WAIR) | 20 | 0 | 1.0 |
| Steam inlet flow (WSTM) | 6.0 | 0 | 1.0 |
| Char extraction (WCHR) | 3.5 | 0 | 0.2 |

Gasifier outputs should be regulated within the limits (table 2) for sink pressure (PSink) disturbance test, load change test and other tests. The desired objective is the outputs should be regulated as closely as possible to the demand.

Table 2. Output limits

| Output variable | Objective | Limits |
|---------------------------------|--------------|------------------------------|
| Fuel Gas Calorific vale (CVGAS) | | ± 10 KJ kg ⁻¹ |
| Bed mass (MASS) | Minimize | ± 500 kg |
| Fuel Gas Pressure (PGAS) | fluctuations | ± 0.1 bar |
| Fuel Gas Temperature (TGAS) | | ± 1 K |

3 Bat Algorithm

BAT algorithm, developed by Xin-She Yang[13], is a population based metaheuristic approach based on hunting behavior of bat. The following idealized rules are assumed for developing code for BAT algorithm [13] [14].

1. All the bats have the ability to identify and locate the prey by echolocation.
2. Bats flies with a frequency f_{min} from the current position x_i at a velocity v_i but with varying loudness and frequency.
3. The loudness varies from a minimum value(A_{min}) to maximum value(A_0)

Figure 1 shows the flow chart for Bat algorithm for the proposed tuning method. The wavelength (λ) and loudness (A_0) of bats varies to search for prey.

The frequency f_i and velocity v_i of i^{th} bat is updated by using the relation

$$f_i = f_{min} + (f_{max} - f_{min})\delta \tag{3}$$

$$v_i^t = v_i^{t-1} + (X_i^t - X_{g_{best}}^t)f_i \tag{4}$$

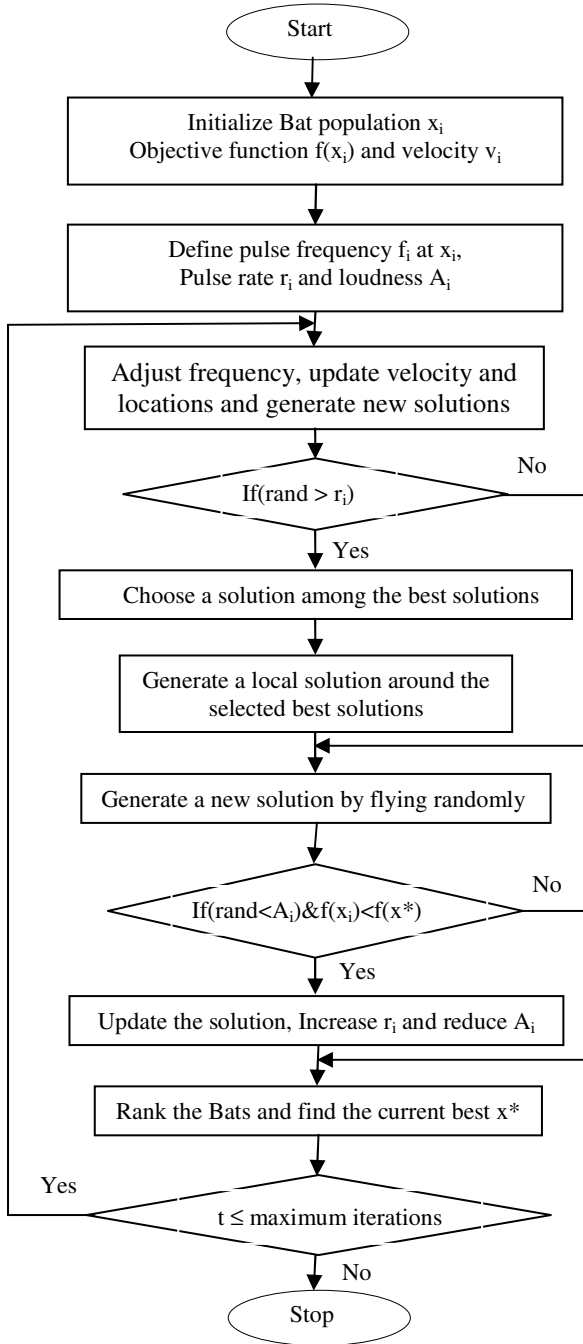


Fig. 1. Flow chart for BAT algorithm

The new solutions can be found by

$$X_i^t = X_i^{t-1} + v_i^t \tag{5}$$

Where,

$\delta[0,1]$ = random vector from a uniform distribution.

f_{min} and f_{max} = function of domain size

Each bat is randomly assigned a frequency between f_{max} and f_{min} . Each bat takes a random walk creating a new solution for itself based on the best current solution given by,

$$X_{new} = X_{old} + \rho A^t \tag{6}$$

Where,

$\rho \in [-1,1]$, a random number and

A^t = average loudness.

Loudness decreases as a bat move closer to its prey and pulse emission rate increases.

$$A_i^{t+1} = \alpha A_i^t \tag{7}$$

$$r_i^{t+1} = r_i^0 [1 - e^{-\gamma t}] \tag{8}$$

Where α and γ are constants.

4 Problem Formulation and Implementation

Figure 2 shows the implementation of BAT algorithm based optimization technique used for tuning the parameters of PI controller of coal gasifier. Maximum of Absolute Error (AE) for PGAS at 0% load and 0% change in coal quality is the objective function for BAT algorithm while controller parameters of pressure loop PI controller is the decision variables.

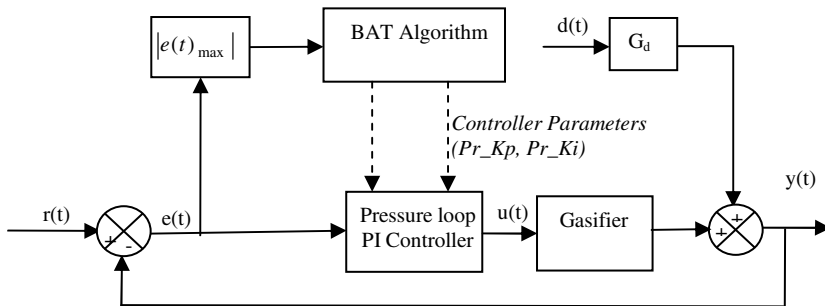


Fig. 2. Block diagram of Optimization scheme

Input constraints are associated with the given Simulink model and it is not included in the specifications. Minimization of the desired performance specifications is the objective function. The controller should respond quickly than the process and hence sampling time is selected as 0.5 seconds.

The procedure is as follows;

- 1) At 0% load and 0% coal quality apply a sinusoidal pressure disturbance (amplitude 0.2bar and frequency of 0.04Hz).
- 2) Run the simulation over 300seconds.
- 3) Calculate maximum Absolute Error (AE).
- 4) Run BAT algorithm (Matlab code).
- 5) Best optimal controller parameters are obtained. These controller parameters of PI controller are the best tuned values for pressure loop PI controller.

Parameters of optimal PI controller and existing PI controller are listed in Table 3. These parameters are used to evaluate the performance of Optimal PI controller.

Table 3. Comparison of PI Controller parameters

| Parameter | Dixon-PI[1] | BAT-PI |
|-----------|----------------|---------------|
| Pr_Kp | 0.00020189 | 0.00032142 |
| Pr_Ki | 2.64565668e-05 | 1.56782164e-7 |

5 Performance Tests

Performance of the gasifier system along with its controller constants is accessed by conducting the following performance tests which include pressure disturbance test, load change test and coal quality test. The optimum controller settings (Pr_Kp and Pr_Ki) for the pressure loop PI controller replace the existing parameters and the following tests are conducted. The gasifier input-outputs should satisfy the constraints (table 1 and table 2) for all performance tests.

5.1 Pressure Disturbance Tests

Change in grid frequency and change in load are the two disturbances that are commonly occurring in a gasifier system. Change in grid frequency causes the inlet valve to move at a low frequency and is represented by sinusoidal pressure disturbance with amplitude of 0.2 bar and a frequency of 0.04Hz. A step disturbance of 0.2 bar represents change in gas turbine valve position due to change in load. Sinusoidal pressure disturbance is applied to gasifier and input-output response is obtained for 300 seconds. Figure 3 shows the input-output response of the gasifier along with its limits. Similar experiment is conducted for step change in pressure and is shown in figure 4. It is clear that input-output response does not violate the constraints as given in table 1 and 2. More particularly at 0% load condition PGAS is well below the limits but with [1] PGAS violates the constraints.

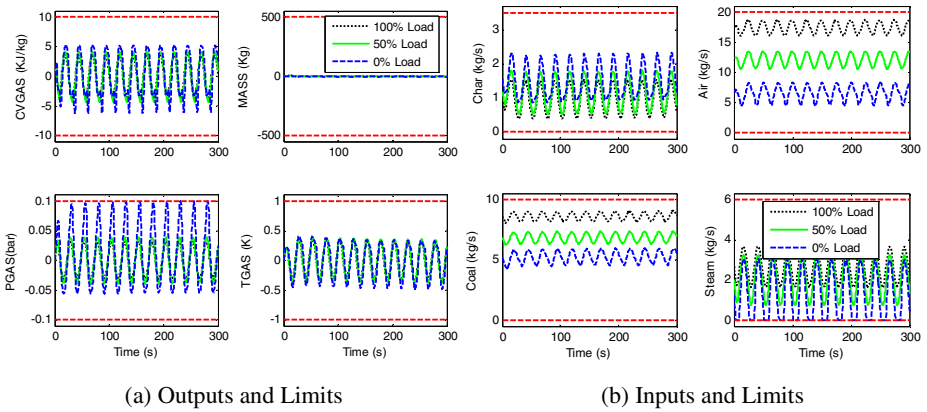


Fig. 3. Response to sinusoidal disturbance at 0%, 50% and 100% load

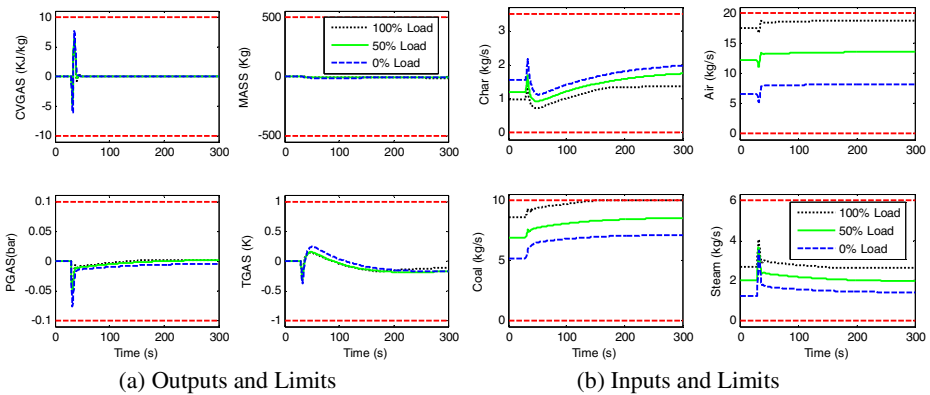


Fig. 4. Response to step disturbance at 0%, 50% and 100% load

The numerical values of above pressure disturbance test are consolidated and are listed in table 4. Test data shows the Integral of Absolute Error (IAE) and maximum Absolute Error (AE) for all the four outputs at all loads and for all pressure disturbances. It is observed that improved response in PGAS while for the other outputs a marginal increase in magnitude is obtained. This is due to the existence of strong interactions among the control loops. The response meets all the performance requirements at all the load conditions (0%, 50% and 100%) and for all the pressure disturbances.

Table 4. Summary of test output results

| Test Description | Output | Maximum Absolute Error | | IAE | |
|---|--------|------------------------|-----------------|-------------------|-------------------|
| | | BAT-PI | Dixon-PI[1] | BAT-PI | Dixon-PI[1] |
| 100% Load, Step Disturbance | CVGAS | 5727.86 | 4885.23 | 66381.33 | 60989.48 |
| | MASS | 6.94 | 6.94 | 1543.32 | 1597.03 |
| | PGAS | 4279.01 | 5018.94 | 158284.93 | 78475.47 |
| | TGAS | 0.27 | 0.24 | 63.25 | 65.09 |
| 50% Load, Step Disturbance | CVGAS | 6283.71 | 5102.16 | 71139.19 | 64766.48 |
| | MASS | 8.45 | 8.45 | 883.41 | 840.04 |
| | PGAS | 5250.58 | 5790.93 | 208911.31 | 94310.73 |
| | TGAS | 0.31 | 0.27 | 74.09 | 77.13 |
| 0% Load, Step Disturbance | CVGAS | 7651.99 | 5875.95 | 89188.03 | 86561.16 |
| | MASS | 11.05 | 11.05 | 1133.53 | 1330.92 |
| | PGAS | 7597.58 | 7714.53 | 469446.96 | 120167.73 |
| | TGAS | 0.38 | 0.32 | 71.12 | 77.05 |
| 100% Load, Sinusoidal Disturbance | CVGAS | 3723.43 | 4101.30 | 1392918.53 | 1545471.04 |
| | MASS | 10.73 | 10.89 | 4153.92 | 4154.65 |
| | PGAS | 3178.29 | 4981.41 | 1188780.64 | 1857629.38 |
| | TGAS | 0.35 | 0.38 | 123.21 | 134.44 |
| 50% Load, Sinusoidal Disturbance | CVGAS | 4298.12 | 4715.68 | 1589982.28 | 1759740.23 |
| | MASS | 12.67 | 12.87 | 5035.52 | 5041.36 |
| | PGAS | 3965.51 | 6209.91 | 1473884.83 | 2307614.42 |
| | TGAS | 0.39 | 0.42 | 136.99 | 149.47 |
| 0% Load, Sinusoidal Disturbance | CVGAS | 6205.03 | 5869.69 | 2050043.67 | 2074977.65 |
| | MASS | 16.26 | 16.35 | 6131.33 | 6016.65 |
| | PGAS | 9948.60 | 11960.42 | 2726992.62 | 3845931.81 |
| | TGAS | 0.48 | 0.48 | 159.54 | 159.09 |

5.2 Load Change Test

Load change test is conducted to verify the stability of the gasifier and controller function across the working range of the plant. The gasifier process is started at 50% load in steady state and ramped it to 100% over a period of 600 seconds (5% per minute). The actual load, CVGAS and PGAS track their demands quickly to setpoint while Bedmass takes more time to reach its steady state, though manipulated inputs coal flow and char flow have reached their steady state immediately (figure 5).

5.3 Coal Quality Test

Carbon content and moisture content of the coal decides the quality of coal gas. Usually the coal quality is not constant over a period of time and may vary to a considerable amount. In this test, the quality of coal increased and decreased by 18% (the maximum possible change in coal quality), and the above pressure disturbance test are conducted to verify the robustness of the controller. Input-output responses

(shown in figure 6-11) for sinusoidal and step change in PSink are verified for 300 seconds. Table 5 shows the violation of the variables under positive (+18%) and negative change(-18%) in coal quality. Since input constraints are inbuilt in the actuator limits, output constraints are considered to be the actual violation. TGAS and PGAS violate the limits under change in coal in coal quality for sinusoidal pressure disturbance and no output variable is found for step pressure disturbance.

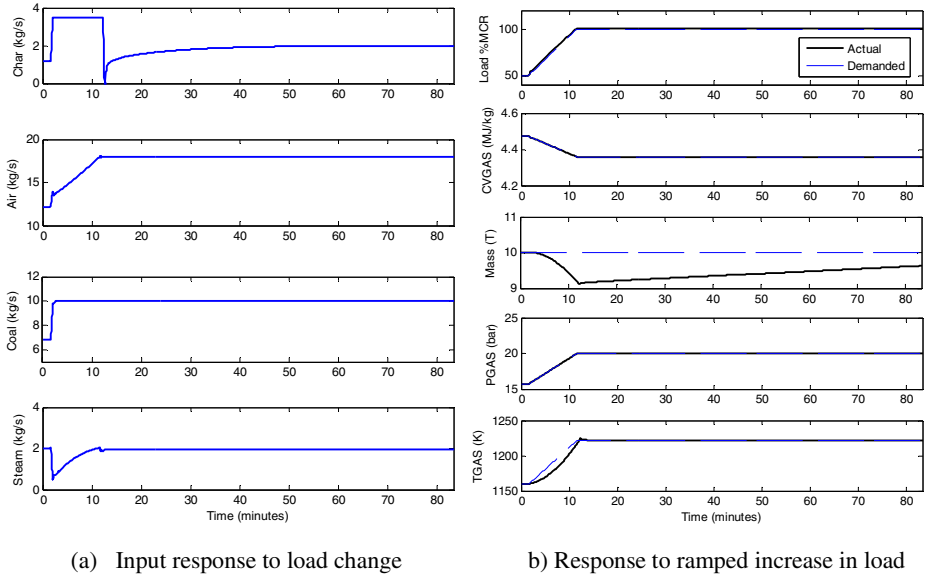


Fig. 5. Response to load increase from 50% to 100% load

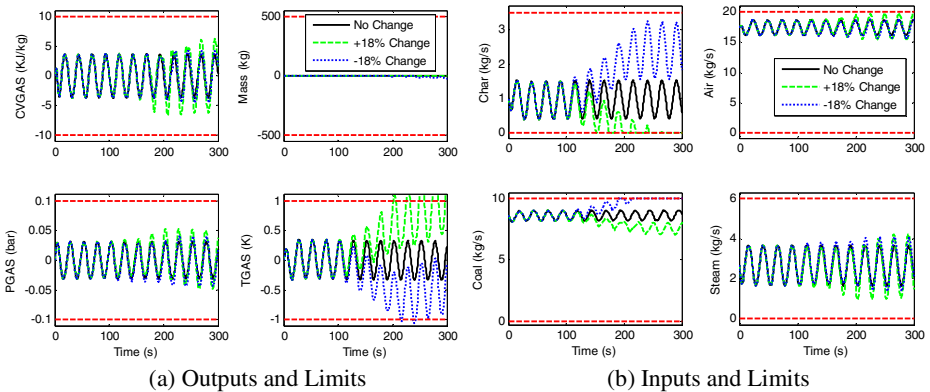


Fig. 6. Response to change in Coal quality at 100 % Load for sinusoidal change in PSink

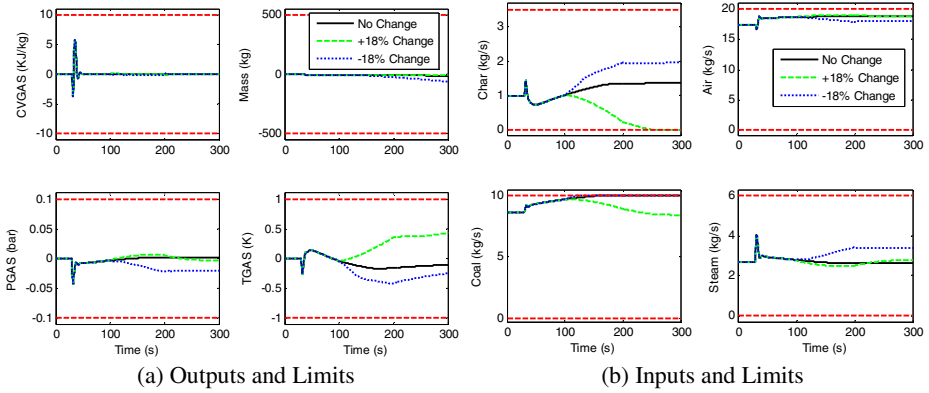


Fig. 7. Response to change in Coal quality at 100 % Load for step change in PSink

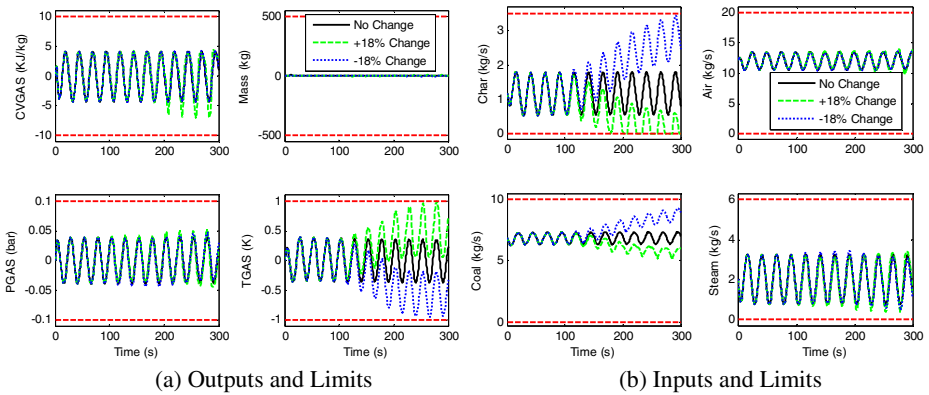


Fig. 8. Response to change in Coal quality at 50% Load for sinusoidal change in PSink

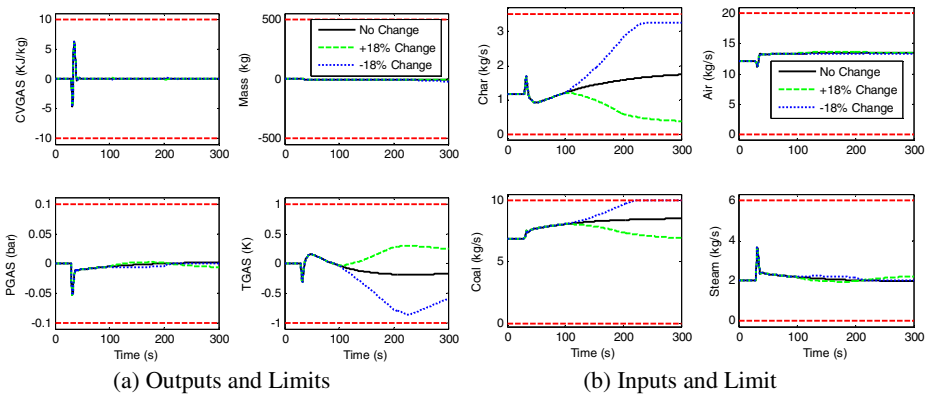


Fig. 9. Response to change in Coal quality at 50 % Load for step change in PSink

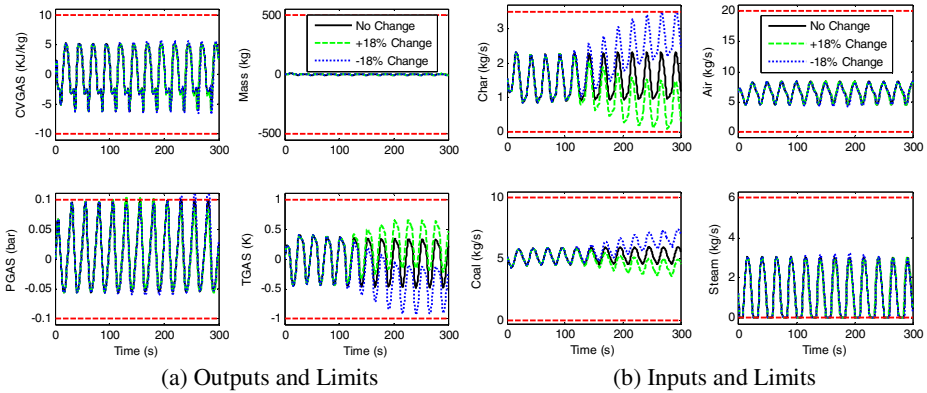


Fig. 10. Response to change in Coal quality at 0 % Load for sinusoidal change in PSink

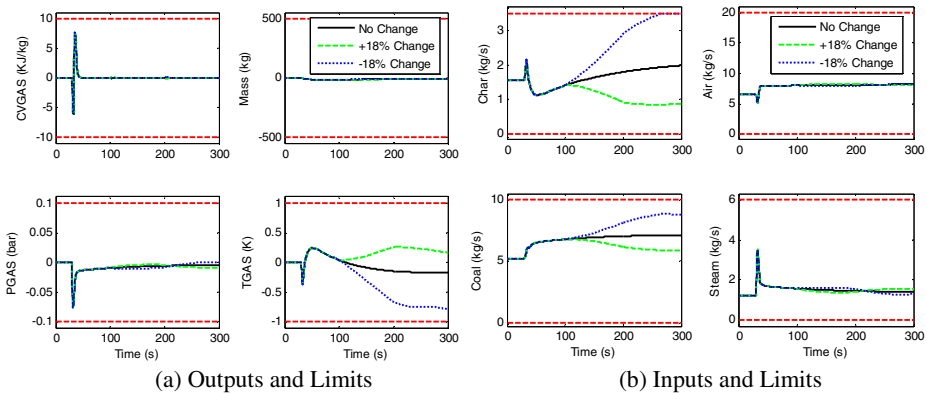


Fig. 11. Response to change in Coal quality at 0% Load for step change in PSink

Table 5. Violation in output variables under coal quality change ($\pm 18\%$) (\uparrow - the variable reaches its upper limit, \downarrow the variable reaches its lower limit)

| Load | 100% | 50% | 0% |
|------------------------------|-------------------|-----------------|-----------------|
| Disturbance type | Sine Step | Sine Step | Sine Step |
| Coal quality increase (+18%) | Tgas \uparrow | Tgas \uparrow | Pgas \uparrow |
| Coal quality decrease (-18%) | Tgas \downarrow | Within limits | Pgas \uparrow |

6 Conclusion

Recent developments in metaheuristic algorithm motivate the authors to find optimum response of coal gasifier under certain constraints. More specifically metaheuristic optimization algorithms are most widely used when the performance objective involves many constraints. In this work the parameters of decentralised PI controller for pressure loop of Coal gasifier is retuned by using BAT algorithm. The existing

controller with tuned parameters does not satisfy the performance requirements at 0% load for sinusoidal pressure disturbance. Best optimum solution for the controller parameters are obtained for the given constraints and these parameters are used to get the desired response and also performance tests are conducted. Performance tests are conducted with the optimum controller settings and the results meet the performance requirements comfortably at 0%, 50% and 100% load conditions.

Acknowledgement. The authors would like to thank Dr. Roger Dixon, Director of Systems Engineering Doctorate Centre, Head of Control Systems Group, Loughborough University, UK for useful communication through email, Xin-She Yang, Senior Research Scientist, National Physical Laboratory, London, UK and the managements of St. Joseph's College of Engineering, Chennai and Sri Krishna College of Engineering & Technology, Coimbatore for their support.

References

1. Dixon, R., Pike, A.: Alstom Benchmark Challenge II on Gasifier Control. IEE Proceedings - Control Theory and Applications 153(3), 254–261 (2006)
2. Chin, C., Munro, N.: Control of the ALSTOM gasifier benchmark problem using H2 methodology. Journal of Process Control 13(8), 759–768 (2003)
3. Al Seyab, R., Cao, Y., Yang, S.: Predictive control for the ALSTOM gasifier problem. IEE Proceedings - Control Theory and Application 153(3), 293–301 (2006)
4. Al Seyab, R., Cao, Y.: Nonlinear model predictive control for the ALSTOM gasifier. Journal of Process Control 16(8), 795–808 (2006)
5. Nobakhti, A., Wang, H.: A simple self-adaptive Differential Evolution algorithm with application on the ALSTOM gasifier. Applied Soft Computing 8(1), 350–370 (2008)
6. Agustriyanto, R., Zhang, J.: Control structure selection for the ALSTOM gasifier benchmark process using GRDG analysis. International Journal of Modelling, Identification and Control 6(2), 126–135 (2009)
7. Tan, W., Lou, G., Liang, L.: Partially decentralized control for ALSTOM gasifier. ISA Transactions 50(3), 397–408 (2011)
8. Huang, C., Li, D., Xue, Y.: Active disturbance rejection control for the ALSTOM gasifier benchmark problem. Control Engineering Practice 21(4), 556–564 (2013)
9. Sivakumar, L., Anitha Mary, X.: A Reduced Order Transfer Function Models for Alstom Gasifier using Genetic Algorithm. Int. J. of Computer Applications 46(5), 31–38 (2012)
10. Kotteeswaran, R., Sivakumar, L.: Lower Order Transfer Function Identification of Nonlinear MIMO System-Alstom Gasifier. International Journal of Engineering Research and Applications 2(4), 1220–1226 (2012)
11. Griffin, I., Schroder, P., Chipperfield, A., Fleming, P.: Multi-objective optimization approach to the ALSTOM gasifier problem. Proc. of IMechE, Part I: Journal of Systems and Control Engineering 214(6), 453–469 (2000)
12. Xue, Y., Li, D., Gao, F.: Multi-objective optimization and selection for the PI control of ALSTOM gasifier problem. Control Engineering Practice 18(1), 67–76 (2010)
13. Yang, X.S.: Bat Algorithm for Multiobjective Optimization. Int. J. Bio-Inspired Computation 3(5), 267–274 (2011)
14. Koffka, K., Sahai, A.: A Comparison of BA, GA, PSO, BP and LM for Training Feed forward Neural Networks in e-Learning Context. Int. J. of Intelligent Systems and Applications 4(7), 23–29 (2012)

Optimal Velocity Requirements for Earth to Venus Mission Using Taboo Evolutionary Programming

M. Mutyalarao^{*}, Amaranathan Sabarinath, and M. Xavier James Raj

Applied Mathematics Division, Vikram Sarabhai Space Centre, ISRO P.O.,
Trivandrum -695022, India
m_mutyalarao@vssc.gov.in

Abstract. In the recent years stochastic optimization techniques have seen a remarkable development for many scientific and industrial problems. There has been growing interest in the successful use of these techniques for obtaining the global optimum of a given optimization problem. Many authors proposed different methods of optimization techniques and applied on a variety of problems. The difficulty in global optimization increases with the dimension of the problem and the presence of multiple local minima. Taboo evolutionary programming (TEP) is a novel evolutionary programming technique, which found extensive usage in the present decade. TEP is an effective global optimization technique which avoids entrapment in local minima and to continue the search to give the optimal solution independent of the initial conditions. The algorithm can be effectively implemented on any complex problem to find the optimum solution with many constraints. Successful use of TEP is published in terms of exploring efficiently the solution space by providing optimal solutions and good initial estimation.

In the present study TEP algorithm is used to find the optimum impulsive velocity requirements for a complex interplanetary trajectory from Earth to Venus mission. The optimization problem is complex due to the presence of the multiple local minima exists in a given synodic cycle. TEP algorithm is integrated with analytic technique called Lambert conic determination. Two types of missions namely flyby and orbiter are considered to explore the optimum trajectory opportunities from elliptical Earth parking orbit. To demonstrate the superiority of TEP, the results were compared with Genetic Algorithm (GA) also.

Keywords: Taboo Evolutionary Programming, Genetic Algorithm, optimum, flyby, orbiter, Lambert conic.

1 Introduction

Interplanetary trajectory design has been a major interest for the space scientific community due to the growing interest in planetary explorations. It is necessary to have the details regarding the impulsive velocity requirements at each maneuver for

^{*} Corresponding author.

minimizing the propellant requirements, while planning advanced missions like flyby, orbiter, Lander, swing by, powered trajectories, etc. Optimization techniques to find best solutions for minimizing the total impulsive velocity requirement (or maximizing the payload capabilities) of interplanetary trajectories have been a subject of considerable importance for several years. Many authors proposed different methods of optimization and implemented on a variety of test cases (Yao et al. (1999), Cvijovic and Klinowski (1995)). Recent years have witnessed an enormous increase in the successful use of stochastic optimization techniques for practical problems in the fields of science, technology, economics, logistics, and travel scheduling etc., which involve global optimization. Many cases the successful use of these techniques is published in terms of exploring, efficiently the solution space by providing optimal solutions and good initial guess. Difficulty in global optimization increases with the dimension of the problem (defined as the number of variable involved in the solution space) and the presence of multiple local minima. Further any effective global optimization technique must be able to avoid entrapment in local minima and continue the search to give the optimal solution independent of the initial conditions. Recently, the authors implemented and analyzed Taboo evolutionary programming (TEP) for computing the optimum transfer trajectory opportunities from Earth to Mars (Mutyalarao et al. (2011)). Numerous unmanned missions between the years 1962 and 1985 to Venus achieved landing on the surface, but all without return. The Launch windows for Venus occur every 19 months (synodic period) and every window was utilized to launch reconnaissance probes. Recent orbiter missions include *Venus Express* of ESA, achieved a polar orbit on April 11, 2006, and it has been sending back science data. *Akatsuki*, was launched on May 20, 2010, by JAXA. Some of the future mission includes, NASA the *Venus In-Situ Explorer*, to be launched in 2013. It would land and perform experiments on the surface of Venus, including taking a core sample and measuring its composition. ESA has proposed the *Venus Entry Probe* to be launched around the same time. Also, the *Venera-D* spacecraft has been proposed by ROSCOSMOS around 2016, and its prime purpose is to map Venus's surface. Recently, Mutyalarao et al (2012) used Lambert conic technique alone for computing minimum energy Earth departure trajectories to Venus. The study deals with the fly-by and orbiter missions to Venus from the Earth parking orbit during the year 2013-14. Grid search with time step of 1 day has been used in their study to compute the optimum. However, the approach requires very high computational time to arrive at the accurate time epoch which gives the minimum energy opportunity.

In the present study an analysis is carried out using Taboo evolutionary programming (TEP) to a complex interplanetary trajectory problem of direct ballistic transfer from Earth to Venus. The methodology is integrated with TEP with Lambert conic technique and it has been applied to find the minimum energy opportunity (or minimum impulsive velocity requirement) transfer trajectories from Earth to Venus for a synodic cycle 2013-2020. The methodology is also integrated with GA and compared with TEP results. The comparison revealed that the results obtained with TEP are better than GA in terms of identifying the global minima.

2 Taboo Evolutionary Programming

2.1 Taboo Search (TS)

In 1995, Taboo (or 'Tabu' being a different spelling of the same word) search (TS), originally developed by Glover (1989, 1990) and extended to continuous valued functions, is a stochastic optimization method attracted much attention. A study with a number of benchmark test examples covering constrained and unconstrained functions was carried out by Rajesh et al (2000) along with a rigorous comparison of the performance of the TS with other methods. The convergence of TS for continuous function optimization is studied by Mingjun Ji and Jacek Klinowski (2006). The results clearly reveal that TS technique can be a viable alternative to other methods such as GA, Simulated annealing based on stochastic differential equations, pure random search etc.

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

2.2 Evolutionary Programming (EP)

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

In 2006, a new method of global optimization technique, Taboo evolutionary programming motivated by combining TS and EP was first introduced by Mingjun Ji & Jacek Klinowski (2006). TEP essentially combines the features of an EP, called single-point mutation (Ji et al. (2004)) with TS. The results were found to be in good agreement with that of analytical results.

2.3 TEP Algorithm

The objective is to compute the minimum of $f(x)$, such that $x \in \Omega$ where $\Omega = \{x \in R^n: a \leq x(j) \leq b, a, b \in R, j = 1, 2, \dots, n\}$, f is a real-valued continuous function on Ω .

The TEP algorithm presented here is closely follows Mingjun Ji & Jacek Klinowski (2006):

- 1) Generate the initial population of μ individuals based on a uniform distribution, and set $k = 1$. Each individual is taken as a vector $x_i, \forall i \in \{1, 2, \dots, \mu\}$
- 2) Evaluate the fitness score for each individual, $x_i, \forall i \in \{1, 2, \dots, \mu\}$, of the population on the objective function, $f(x_i)$.
- 3) For each parent $x_i, i \in \{1, 2, \dots, \mu\}$, create a single offspring x'_i by

$$x'_i(j_i) = x_i(j_i) + \eta N_i(0,1), \quad \eta = \eta \exp(-\alpha),$$

where j_i is randomly chosen from the set $\{1, 2, \dots, n\}$ and the other components of x'_i are equal to the corresponding x_i 's. $N(0,1)$ denotes a normally distributed one-dimensional random number with a mean of zero and a standard deviation of one. Here, the parameter $\alpha = 1.01$. The initial value of η is $\frac{b-a}{2}$ and whenever $\eta < 10^{-4}$ then η is set to its initial value.

- 4) Calculate the fitness of each offspring $x'_i, \forall i \in \{1, 2, \dots, \mu\}$.
- 5) Perform the search using the following improved paths:
 - 5.1 Choose an improved path as follows. For each $i \in \{1, 2, \dots, \mu\}$, if $f(x'_i) \leq f(x_i)$ then $y_i = x'_i, d_i = x'_i - x_i$ is an improved path. Put a pair of vectors (y_i, d_i) into the set A .
 - 5.2 Choose r best fitness individuals from the set A as the parents with improved paths. Note that $(y_m, d_m), m = 1, 2, \dots, r$, where y_m is an objective variable and d_m is the corresponding improved path. Set $A = \emptyset$, where \emptyset is a null set.
 - 5.3 Calculate fitness: for each $m = 1, 2, \dots, r, y'_m = y_m + \rho d_m, \rho = \rho \exp(-\alpha)$. The initial value of ρ is 1 and whenever $\rho < 10^{-6}$, then ρ is set to its initial value.
 - 5.4 For each $m = 1, 2, \dots, r$, if $f(y'_m) \leq f(y_m)$, then set (y'_m, d_m) as a parent of the next generation with improved search paths, and put into the set A .
 - 5.5 Record the number τ , of members in set A .
- 6) Choose the parents for the next generation.
 - 6.1 Perform a comparison over the union of parents x_i and offspring $x'_i, \forall i \in \{1, 2, \dots, \mu\}$. For each individual, q opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individuals fitness is equal to or greater than the opponent's, it scores a 'win'. Select the $\mu - \tau$ individuals out of x_i and $x'_i, \forall i \in \{1, 2, \dots, \mu\}$, which have the most wins to be put into the set B .
 - 6.2 Make the individuals x, x' and y from sets B and A the parents of the next generation. Set $B = \emptyset$.
- 7) Check the Taboo status as follows:
 - 7.1 Record the current optimal fitness, f_k^* , and the current optimal solution, x_k^* .
 - 7.2 When $k > L$, and for a specified σ_1 and σ_2 (sufficiently both are small real numbers) compare the optimal fitness of the current generation with the optimal

fitness of the previous L generations. Thus, if $|f_k^* - f_{k-L}^*| \leq \sigma_1$, then $f^* = f_k^*$, $x^* = x_k^*$. Put the pair of vectors (f^*, x^*) into the taboo table ψ . The length of taboo table ψ is l .

7.3 For any (f^*, x^*) in ψ , if the current optimal fitness f_k^* and the optimal solution x_k^* satisfies the taboo conditions, $|f_k^* - f^*| \leq \sigma_1$ and $\|x_k^* - x^*\| \leq \sigma_2$, then generate the initial population of μ individuals and set new individuals as the k^{th} generation.

8) Terminate if k is more than maximum number of generations. Otherwise, set $k = k + 1$ and go to Step-3.

3 Problem Formulation

The objective function considered here is the total velocity requirements (ΔV) at the end of a direct ballistic transfer from Earth to Venus. The optimization includes the minimization of ΔV under the constraints of co-orbital plane maneuvers. The search space is characterized by three design variables. viz., departures date (DD), transfer duration (Δt), and the orbital inclination (i).

4 Methodology

As indicated in Cornelisse (1978) for an interplanetary mission, a spacecraft is launched from Earth and it is accelerated to a velocity greater than the local escape velocity (V_e), the spacecraft recedes from Earth along a hyperbolic trajectory (planeto-centric). As distance from Earth increases, solar attraction gradually becomes more important until finally the spacecraft enters into a helio-centric trajectory. The spacecraft fly along this trajectory to reach the neighborhood of the target planet Venus, where the latter's gravitational attraction of Venus gradually overtakes the solar gravitational attraction and it enters into a hyperbolic trajectory about the target planet Venus. If no orbital maneuver is executed during this phase, the spacecraft will pass the target planet Venus and recedes from it again along the outgoing leg of the hyperbola. This constitutes a fly-by or swing-by mission. For an orbiter mission, the spacecraft is decelerated to enter into a closed orbit about the target planet Venus; for a lander mission, the spacecraft's velocity with respect to the surface of the planet must be reduced to almost zero. The Earth to Venus transfer trajectory design methodology considered in this study is based on direct ballistic technique, for which the input requirements are the date of Earth departure date (DD), time of flight (Δt) and inclination (i) of the Earth parking orbit. Earth and Venus are assumed to be point masses. The planetary ephemeris in J2000 reference frame is generated using the analytical expressions available in Standish et al. (1992) for mean orbital elements

relative to the ecliptic plane. The methodology used in this paper is presented in Figure 1. The computation of ΔV in Figure 1 is given below:

- Compute the state vector (\vec{r}_1, \vec{v}_1) of Earth at the departure date (DD).
- Compute the state vector (\vec{r}_2, \vec{v}_2) of Venus at the arrival date (i.e., DD+ Δt).

The planetary ephemerides were modeled using Meeus algorithm [11].

- Use Lambert problem solution technique (universal variable method) [12, 13] for the departure to target phase and determine initial (\vec{v}_i) and final (\vec{v}_f) vectors of the transfer hyperbola.

- Compute the asymptotic relative velocity vector at departure $(\vec{v}_{\infty D})$ and at the arrival $(\vec{v}_{\infty A})$ by using the following formulae:

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

- Transform $\vec{v}_{\infty D}$ to Earth Centered Inertial frame and calculate the right ascension and declination to identify the co-orbital transfer trajectory by using orbital inclination (i).

- Compute the impulsive ΔV as follows:

A. For Fly-by mission

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

$$v_h = \sqrt{\frac{2\mu_E}{r_p} + v_{\infty D}^2} \tag{1}$$

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

$$v_p = \sqrt{\frac{2\mu_E}{r_p} - \frac{2\mu_E}{r_p+r_a}} \tag{2}$$

Hence the departure velocity required is

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

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

B. For orbiter mission

Here the total velocity required ΔV is

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

where, ΔV_1 is the departure impulse calculated from equations (1) to (3) and ΔV_2 is the arrival impulse. ΔV_2 can be calculated by replacing μ_E , $v_{\infty D}^2$, r_p and r_a with the suitable constraints at arrival phase of the planet Venus in equations (1) to (3). The constants used for the present study is provided in [15].

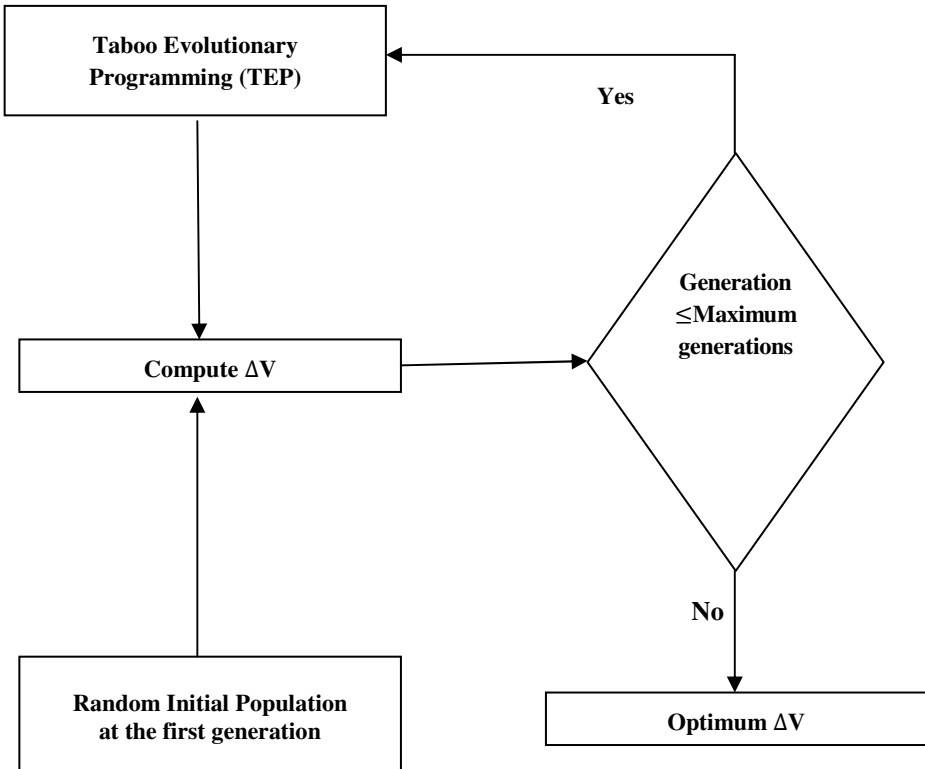


Fig. 1. Schematic diagram of methodology

5 Results

Computation of ΔV mentioned in section 4.1 is validated in Mutyalarao et al.(2012) for Earth to Venus transfer with a test case given in Stephen Kemble (2006). In the present study, Earth to Venus transfer is considered from an initial Earth parking orbit of 250×23000 km with 18° and 50° inclinations for all the 5 opportunities during the synodic cycle 2013-2020. In order to find the global minimum energy opportunity for fly-by mission to Venus, minimization of departure impulsive ΔV_1 is considered as an objective function. For orbiter missions, 1000 km circular orbit is considered around Venus. The total impulsive ΔV calculated from equation (4) is minimized for orbiter mission. Figures 2 and 3 give the contour charts of departure ΔV_1 for fly-by and total ΔV for orbiter missions, respectively for the year 2013-14 with earth parking orbit inclination of 18° . The appropriate launch window can be identified from these contour plots.

TEP Simulations are performed for all five opportunities spanning one synodic cycle i.e., 2013-2020. To evaluate the performance, the algorithm was run for 50

times with 100 generations each. The convergence plots for fly-by and orbiter missions for one typical run, executed for the year 2015, are given in Figures 4 and 5, respectively. From these figures, it can be seen that the run is converged to identify the global optimum within 20 generations. Statistics of velocity dispersions over 50 simulations for flyby mission in one Synodic cycle for 18° and 50° inclinations are tabulated in Tables 1 and 2, respectively. The same details for orbiter mission are provided in Tables 3 and 4. From these Tables, it is evident that the standard deviation (σ) for all the cases is less than 75 m/s only. Similarly the difference between maximum and minimum values of ΔV is also less for all the opportunities. For flyby mission opportunities, ΔV requirement during 2013-16 varies between 1.333 km/s and 1.521 km/s and for orbiter mission it is between 4.741 km/s and 5.607 km/s.

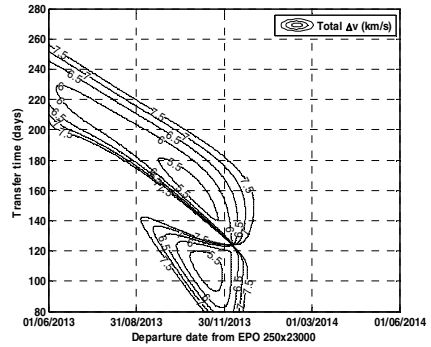
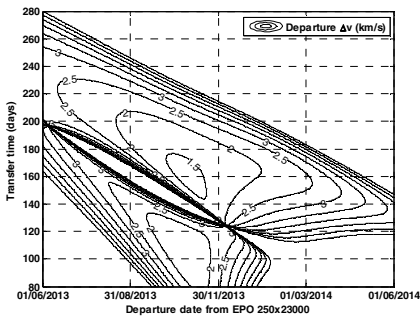


Fig. 2. Departure ΔV_1 contours for fly-by mission in 2013-14 with 18° inclination

Fig. 3. Total ΔV contours for orbiter mission in 2013-14 with 18° inclination

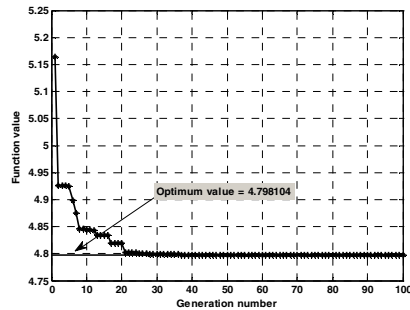
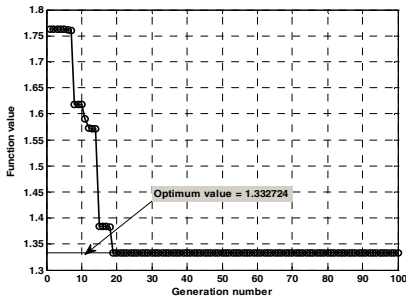


Fig. 4. Flyby mission convergence plot (Departure Epoch: 20/05/2015 11:51:21)

Fig. 5. Orbiter mission convergence plot (Departure Epoch: 07/06/2015 16:27:44)

Table 1. Optimum launch opportunity for fly-by mission during a synodic cycle for $i=18^\circ$

| No | Departure Epoch* | Δt (days) | ΔV (km/s) | Statistics of ΔV (km/s) from 50 runs | | | |
|----|---------------------|-------------------|-------------------|---|----------|----------|----------|
| | | | | Mean | σ | Min | Max |
| 1 | 02/11/2013 04:11:09 | 160.426 | 1.416028 | 1.430946 | 0.020913 | 1.416028 | 1.532951 |
| 2 | 20/05/2015 11:51:21 | 158.262 | 1.332724 | 1.341879 | 0.029386 | 1.332724 | 1.458983 |
| 3 | 28/12/2016 19:48:30 | 132.712 | 1.378857 | 1.410366 | 0.042688 | 1.378857 | 1.495838 |
| 4 | 31/07/2018 18:36:16 | 122.655 | 1.393994 | 1.413369 | 0.023845 | 1.393994 | 1.552093 |
| 5 | 10/04/2020 21:49:23 | 173.514 | 1.520652 | 1.547792 | 0.021723 | 1.520652 | 1.569358 |

* Date/Month/Year Hours:Minute:Second

Table 2. Optimum launch opportunity for fly-by mission during a synodic cycle for $i=50^\circ$

| No | Departure Epoch* | Δt (days) | ΔV (km/s) | Statistics of ΔV (km/s) from 50 runs | | | |
|----|---------------------|-------------------|-------------------|---|----------|-----------|-----------|
| | | | | Mean | σ | Min | Max |
| 1 | 30/10/2013 18:34:46 | 158.811 | 1.408716 | 1.408716 | 1.0E-06 | 1.408716 | 1.408719 |
| 2 | 20/05/2015 12:11:27 | 158.250 | 1.332724 | 1.335355 | 0.015980 | 1.332724 | 1.445945 |
| 3 | 28/12/2016 19:51:54 | 132.709 | 1.378857 | 1.381391 | 0.013281 | 1.378857 | 1.471657 |
| 4 | 31/07/2018 18:49:31 | 122.646 | 1.393994 | 1.395464 | 0.005690 | 1.393994 | 1.424096 |
| 5 | 27/03/2020 13:49:32 | 171.659 | 1.462568 | 1.462568 | 3.33E-08 | 1.4625680 | 1.4625683 |

Table 3. Optimum launch opportunity for orbiter mission during a synodic cycle for $i=18^\circ$

| No | Departure Epoch | Δt (days) | ΔV (km/s) | Statistics of ΔV (km/s) from 50 runs | | | |
|----|---------------------|-------------------|-------------------|---|----------|----------|----------|
| | | | | Mean | σ | Min | Max |
| 1 | 13/11/2013 01:47:30 | 108.969 | 5.213998 | 5.222229 | 0.032937 | 5.213998 | 5.359453 |
| 2 | 07/06/2015 16:27:44 | 130.735 | 4.798104 | 4.807720 | 0.015716 | 4.798104 | 4.859871 |
| 3 | 02/12/2016 19:14:40 | 160.206 | 4.800831 | 4.889158 | 0.050440 | 4.800831 | 4.987421 |
| 4 | 18/05/2018 23:49:59 | 194.634 | 5.259231 | 5.339707 | 0.044824 | 5.259231 | 5.448309 |
| 5 | 01/04/2020 13:28:51 | 109.275 | 5.606557 | 5.606589 | 0.000180 | 5.606557 | 5.607886 |

Table 4. Optimum launch opportunity for orbiter mission during a synodic cycle for $i=50^\circ$

| No | Departure Epoch* | Δt (days) | ΔV (km/s) | Statistics of ΔV (km/s) from 50 runs | | | |
|----|---------------------|-------------------|-------------------|---|----------|----------|----------|
| | | | | Mean | σ | Min | Max |
| 1 | 13/11/2013 01:47:34 | 108.969 | 5.213998 | 5.247404 | 0.043118 | 5.213998 | 5.308383 |
| 2 | 07/06/2015 16:27:43 | 130.735 | 4.798104 | 4.803389 | 0.011160 | 4.798104 | 4.846295 |
| 3 | 06/12/2016 05:21:00 | 161.593 | 4.741006 | 4.747886 | 0.034051 | 4.741006 | 4.916637 |
| 4 | 10/06/2018 15:33:07 | 183.917 | 5.011725 | 5.025553 | 0.049337 | 5.011725 | 5.357087 |
| 5 | 08/01/2020 11:51:24 | 197.981 | 5.446010 | 5.518154 | 0.074642 | 5.446010 | 5.606560 |

In order to find out the effectiveness of TEP, the results were compared with GA for 18° inclination cases. The population size is selected to be 100. Because all optimization parameters have a specified range, a binary coded GA is utilized and all parameters are coded in 60 bits. A single point crossover with probability of 0.8, and bit-wise mutation with probability of 0.01 are selected. The GA is run for 100 generations. Table 5 shows the results obtained from 50 runs using TEP and GA. From this Table, it is evident that the standard deviation (σ) is less for all cases in TEP. Similarly the difference between maximum and minimum values of ΔV is also less in TEP. Further, it is noticed that from the numerical results that the computational time is significantly less for TEP comparing to GA.

Table 5. Comparison of ΔV and computational time for 50 simulations

| Optimum ΔV (km/s) | | | | |
|------------------------------|------------------------|--------------|--------------------------------------|--------------|
| | Genetic Algorithm (GA) | | Taboo Evolutionary Programming (TEP) | |
| | Case-1 | Case-2 | Case-1 | Case-2 |
| | Fly-by 2016 | Orbiter 2018 | Fly-by 2016 | Orbiter 2018 |
| Mean | 1.497071 | 5.340347 | 1.410366 | 5.339707 |
| σ | 0.049734 | 0.082486 | 0.042688 | 0.044824 |
| Minimum | 1.378911 | 5.260623 | 1.378857 | 5.259231 |
| Maximum | 1.589579 | 5.587782 | 1.495838 | 5.448309 |
| Computational time (seconds) | | | | |
| | Genetic Algorithm (GA) | | Taboo Evolutionary Programming (TEP) | |
| | Case-1 | Case-2 | Case-1 | Case-2 |
| | Fly-by 2016 | Orbiter 2018 | Fly-by 2016 | Orbiter 2018 |
| Mean | 6.925891 | 6.310686 | 5.2449866 | 5.4564853 |
| σ | 1.822184 | 1.410254 | 0.593859 | 0.9678942 |
| Minimum | 5.484251 | 5.360424 | 3.3467888 | 3.3568422 |
| Maximum | 10.849716 | 10.773232 | 10.45668965 | 9.4454789 |

6 Conclusions

In this paper the detailed algorithm for minimum energy opportunities for direct transfer trajectories from Earth to Venus using Taboo Evolutionary Programming is presented. The results for fly-by and orbiter missions are generated by integrating the Taboo Evolutionary Programming. The results are generated for one synodic cycle 2013-2020 by minimizing the total ΔV requirement. This algorithm has been successful in identifying the global minimum of the total ΔV requirement in all the five opportunities in one cycle. The results of TEP are compared with GA and it shows a substantial improvement in identifying the global optimum.

References

1. Battin, R.H.: An introduction to the mathematics and methods of astrodynamics, AIAA education series (1987)
2. Cornelisse, J.W.: Trajectory Analysis for Interplanetary Mission. *ESA Journal* 2, 131–164 (1978)
3. Cvijovic, D., Klinowski, J.: Taboo search: an approach to the multiple minima problem. *Science* 267, 664–666 (1995)
4. Glover, F.: Tabu search- part I. *ORSA Journal of Computing* 1, 190–206 (1989)
5. Glover, F.: Tabu search-part II. *ORSA Journal of Computing* 2, 4–32 (1990)
6. Ji, M., Tang, H., Guo, J.: A single-point mutation evolutionary programming. *Inf. Process.Lett.* 90, 293–299 (2004)
7. Ji, M., Klinowski, J.: Taboo evolutionary programming: a new method of global optimization. *Proc. R. Soc. A* 462, 3613–3627 (2006)
8. Ji, M., Klinowski, J.: Convergence of taboo searches in continuous global optimization. *Proc. R. Soc. A* 462, 2077–2084 (2006)
9. Mutyalarao, M., Sabarinath, A., Raj, M.X.J.: Taboo Evolutionary Programming Approach to Optimal Transfer from Earth to Mars. In: Panigrahi, B.K., Suganthan, P.N., Das, S., Satapathy, S.C. (eds.) SEMCCO 2011, Part II. LNCS, vol. 7077, pp. 122–131. Springer, Heidelberg (2011)
10. Rajesh, J., Jayaraman, V.K., Kulkarni, B.D.: Taboo search algorithm for continuous function optimization. *Trans. IChemE* 78, Part A (September 2000)
11. Standish, E.M., Newhall, X.X., Williams, J.G., Yeomans, D.K.: Orbital Ephemerides of the Sun, Moon and Planets. In: Seidelmann, P.K. (ed.) *Explanatory Supplement to the Astronomical Almanac*. University Science Books (1992)
12. Kemble, S.: *Interplanetary Mission Analysis and Design*. Springer, Heidelberg (2006)
13. Vallado, D.A.: *Fundamentals of astrodynamics and applications*, 2nd edn. Kluwer Academic Publishers, London (2001)
14. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. *IEEE Trans. Evol. Comput.* 3, 82–102 (1999)
15. Mutyalarao, M., Raj, M.X.J.: Minimum energy Earth departure trajectories to Venus. In: *Proceedings of the Fifty Seventh Congress of the Indian Society of Theoretical and Applied Mechanics (An International meet)*, pp. 109–114 (2012)

Author Index

- Abdelaziz, Almoataz Y. I-424, I-504
Abdelsalam, H.A. I-504
Abdul Kadhar, K. Mohaideen II-32
Abhyankar, A.R. I-209
Agrawal, Sanjay I-88
Ahmed, Sumaiya I-68
Akella Venkata, Bharadwaj I-366
Anavatti, Sreenatha G. II-476
Anbazhagan, S. II-499
Anumandla, Kiran Kumar I-366
Aruna, M. II-44
Arunachalam, Sundaram I-354
Arya, K.V. I-248
Asaithambi, Mythili II-523
Athaide, Joanne II-364
- Babulal, C.K. I-321, I-710
Bajer, Dražen I-158
Bakkiyaraj, R. Ashok I-580
Bali, Shivani II-546
Banerjee, Abhik II-119
Banerjee, Tribeni Prasad II-574
Bansal, Jagdish Chand I-248
Baskar, S. II-32, II-511
Baskar, Subramanian I-47, I-146, I-481
Basu, Debabrota I-59, I-222, I-738
Bathrinath, Sankaranarayanan I-377
Behera, Laxmidhar II-442
Benala, Tirimula Rao II-205
Bhattacharjee, Anup Kumar I-436
Bhattacharya, Bidishna I-78
Bhattacharyya, Dhruba Kumar II-95
Bhattacharyya, Saugat II-534
Bhuvaneshwari, S. II-44
Biswal, Bibhuti Bhusan II-277
Biswas, Animesh II-406
Biswas, Subhodip I-222
Biswas (Raha), Syamasree I-345
Burman, Ritambhar I-274
Buyukdagli, Ozge I-1, I-24
- Cao, Cen I-688
Cernea, Ana I-642
Chakraborty, Aruna II-354
Chakraborty, Niladri I-78, I-133, I-189, I-345
Chatterjee, Sarthak I-286
Chaturvedi, D.K. II-132
Chaudhri, D.S. II-617
Chaudhari, Devendra S. II-156
Chaudhuri, Sheli Sinha I-179, I-222
Chawla, Akshay I-13
Chowdhury, Archana II-55
- Das, Asit Kumar II-144
Das, Sanjoy II-487
Das, Shantanu I-457
Das, Swagatam I-179, I-222, I-236, I-260, I-274, I-286, I-298, I-310, I-401, I-457, I-469, I-515, II-487, II-574
Dasgupta, Preetam I-515
Dash, S.S. II-639
De, Arunava I-436
Deb, Kalyanmoy I-13
Debchoudhury, Shantanab I-59, I-738
Deep, Aakash I-199
Deepamangai, P. I-662
Deepika, T.J. I-617
Dehuri, Satchidananda II-205
Devaraj, D. II-397
Devesh Raj, Mani I-699
Devi, S. I-673
Dewan, Hrishikesh I-630, I-725
Dhal, P.K. II-342
Dhingra, Atul II-248
Dora, Lingraj I-88
Du, Huimin I-688
Dubey, Hari Mohan I-568
Duraipandy, P. II-397
Dutta, Malayananda II-95
- Elango, Murugappan II-14
El-Fergany, Attia I-424
- Fernández-Martínez, Juan Luis I-642
- Ganguly, Srinjoy I-401
Gao, Kai-Zhou I-59, I-738

- Gao, X.Z. II-625
 García-Gonzalo, Esperanza I-642
 Garg, Akhil II-23
 Gauthaam, M. II-658
 Geethanjali, M. I-673
 Ghosh, Arka II-144
 Ghoshal, Sakti Prasad I-35, I-98, I-547, I-558
 Ghoshal, S.P. II-119
 Goswami, Debdipta I-286
 Goswami, Rajib II-95
 Gunasundari, Selvaraj II-214
- Hanmandlu, Madasu II-248
 Hassanein, Osama II-287
 Hei, Xinhong I-389
- Indulkar, C.S. II-237
 Islam, R. I-547
- Jadon, Shimpi Singh I-248
 Jaipuria, Sanjita II-69
 Jana, Nanda Dulal II-193
 Janakiraman, S. II-214
 Janarthanan, Ramadoss II-55, II-354, II-376, II-534
 Jauhar, Sunil Kumar I-199
 Jegatheesan, R. I-617
 Jha, Panchanand C. II-277, II-546
 Jiang, Qiaoyong I-389
 Joe Amali, S. Miruna I-146
- Kabat, Manas Ranjan I-68, I-590
 Kanagaraj, Ganesan II-14
 Kandaswamy, A. II-658
 Kannan, B.K.V. I-377
 Kapoor, Neha II-648
 Kar, Rajib I-35, I-98, I-547
 Kar, Reshma II-354
 Karabulut, Korhan I-1
 Karthikeyan, P. I-481
 Kaur, Ramandeep II-546
 Kesavadas, C. II-268
 Kim, Dong Hwa II-625
 Kiran, Deep I-209
 Kirsanov, Andrey II-476
 Kiziay, Damla I-1
 Kolte, Mahesh T. II-156, II-617
 Konar, Amit II-55, II-354, II-376, II-534
 Kotteswaran, Rangasamy I-333, I-750
 Krishnanand, K.R. II-107
- Krishnaswamy, S. II-511
 Kudikala, Shravan I-366
 Kulkarni, Pravin II-595
 Kumar, Amioy II-248
 Kumar, Avanish II-442
 Kumar De, Arnab II-406
 Kumarappan, N. II-499
 Kumarappan, Narayanan I-580, II-499
 Kumari, R. Sheela II-268
 Kundu, Rupam I-310, II-487
 Kundu, Souvik I-222
- Laha, Dipak II-308
 Lozano, Jose A. II-1, II-419
- Madan, Sushila II-546
 Mahapatra, S.S. II-69
 Majhi, Babita II-298
 Maji, Pradipta II-387
 Majumder, Bodhisattwa Prasad I-260
 Malar, E. II-658
 Malik, L.G. I-492
 Mall, Rajib II-205
 Mallipeddi, Rammohan I-170
 Manadal, Kamal Krishna I-345
 Mandal, Durbadal I-35, I-98, I-547
 Mandal, Kamal K. I-78, I-189
 Manic, K. Suresh I-110
 Manoharan, P.S. I-662
 Manoharan, Sujatha C. II-523
 Maroosi, Ali II-257
 Martinović, Goran I-158
 Mathuranath, P.S. II-268
 Meena, Mukesh II-319
 Mendiburu, Alexander II-1, II-419
 Mini, S. I-446
 Mishra, Ambika Prasad II-298
 Mishra, Laxmi Prasad II-332
 Mohanty, Bibhuprasad II-465
 Mohanty, Mihir Narayan II-332, II-465
 Mohanty, Prases Kumar I-527
 Moirangthem, Joy mala II-107
 Mukherjee, Rohan I-310, II-487
 Mukherjee, Satrajit I-260
 Mukherjee, Sudipto I-286
 Mukherjee, V. I-558, II-119
 Muniyandi, Ravie Chandren II-257
 Murthy, V. Venkata Ramana II-85
 Mutyalara, M. I-762

- Nag, Soumyadeep II-431
 Naik, Anima II-180
 Natarajan, V. II-606
 Nayak, Prasanta Kumar II-332
 Nayak, Raksha B. I-630, I-725
 Ni, Qingjian I-688

 Ohri, Jyoti II-648
 Osama, R.A. I-504

 Padole, Chandrashekhar II-364
 Pal, D. I-189
 Pal, Kunal I-298
 Pal, Monalisa II-376
 Pan, Quan-Ke I-24
 Panda, Bishnupriya II-298
 Panda, Rutuparna I-88
 Panda, Sanjib Kumar II-107
 Panda, Sidhartha I-537
 Pandey, Mayank II-559
 Pandiarajan, K. I-321
 Pandit, Manjaree I-568
 Panigrahi, Bijaya Ketan I-209, I-424,
 I-504, I-515, I-568, I-642, II-107, II-248,
 II-487
 Pant, Millie I-199
 Parhi, Dayal R. I-527
 Park, Jin Ill II-625
 Parvathi, K. II-180
 Patel, Manoj Kumar I-68
 Patel, Rahila I-492
 Pati, Soumen Kumar II-144
 Patil, Chandrashekhar G. II-156
 Patra, Gyana Ranjan I-179
 Pattnaik, Laksha II-465
 Paul, Sujoy I-274
 Paul, Sushmita II-387
 Philip, Namitha II-431
 Piplai, Aritran I-260
 Ponnambalam, S.G. I-377, II-14
 Pradhan, Rohini I-590
 Praveena, Pillala II-583
 Premdayal, Sinha Anand II-132

 Raghuwanshi, M.M. I-492
 Raj, M. Xavier James I-762
 Raja, Nadaradjane Sri Madhava I-110
 Rajasekhar, Anguluri I-457, I-469
 Rajinikanth, V. I-110

 Rakshit, Pratyusha II-55, II-534
 Ram, Gopi I-35
 Ramalakshmi, A.P.S. I-662
 Ramalingam, K. II-237
 Ranjan, C. Christober Asir II-342
 Ravi, Vadlamani II-559
 Ravindran, P. II-639
 Ray, Tapabrata II-287, II-476
 Reddy Busireddy, Nagarjuna II-453
 Reddy, M. Umamaheswara I-617
 Rout, Minakhi II-298
 Rout, Umesh Kumar I-537
 Roy, Subhrajit I-236

 Sabarinath, Amaranathan I-762
 Sabarinath, Pandurengan I-601
 Sabat, Samrat L. I-366, I-446
 Sabura Banu, U. I-413
 Saha, Chiranjib I-298
 Saha, Sriparna II-376
 Saha, Suman K. I-98
 Sahoo, Satya Prakash I-68, I-590
 Sahu, Badrinarayan II-332
 Sahu, Rabindra Kumar I-537
 Sangeetha, N. I-354
 Sankar, S. Saravana I-377
 Santana, Roberto II-1, II-419
 Saranya, R. I-354
 Saravanan, R. I-601
 Sardeshmukh, M.M. II-617
 Sarkar, Soham I-179
 Satapathy, Suresh Chandra II-180
 Sattianadan, D. I-652, II-639
 Sen, Aritra I-236
 Sharma, Harish I-248
 Shaw, Binod I-558
 Shukla, Anupam II-168
 Shunmugalatha, A. I-122
 Si, Tapas I-436
 Sil, Jaya II-193
 Singh, Alok II-85
 Singh, N. Albert II-268
 Singh, Pramod Kumar II-319
 Singh, Raj Mohan II-226
 Sivakamasundari, J. II-606
 Sivakumar, Lingappan I-333, I-750
 Som, Trina I-133
 Somasundaram, Periyasami I-699
 Sreenatha, G. II-287
 Srivastava, Divya II-226

- Srivastava, Soumil I-13
 Subramanian, Srinivasan II-523
 Suchitra, D. I-617
 Sudana Rao, N. Madhu II-44
 Sudha, S. II-511
 Sudhakaran, M. II-639
 Suganthan, Ponnuthurai Nagaratnam
 I-24, I-59, I-170, I-738
 Suganyadevi, M.V. I-710
 Sujatha, Kothapalli Naga II-583
 Sur, Chiranjib II-168
 Suresh Babu, B. I-122
 Susheela Devi, V. I-630, I-725
- Tai, Kang II-23
 Tamilselvi, Selvaraj I-47
 Tasgetiren, M. Fatih I-1, I-24
 Thansekhar, M.R. I-601
- Tibarewala, D.N. II-534
 Tonge, Vanita G. II-595
- Udgata, Siba K. I-446, II-453
 Udgir, Mugdha I-568
- Vaisakh, Kanchapogu II-583
 Varghese, Tinu II-268
 Velasco, Julian I-642
 Vidyasagar, S. I-652
 Vijayakumar, K. I-652, II-639
- Wang, Lei I-389
- Yadav, Ajay Pratap II-442
 Yang, Dongdong I-389
- Zou, Feng I-389