ADVANCES IN INTELLIGENT AND *SOFT COMPUTING* 58

Jörn Mehnen
Mario Köppen
Ashraf Saad
Ashutosh Tiwari (Eds.)

Applications of Soft Computing

From Theory to Praxis

Springer

Advances in Intelligent and Soft Computing 58

Editor-in-Chief: J. Kacprzyk

Advances in Intelligent and Soft Computing

Editor-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 43. K.M. Węgrzyn-Wolska, P.S. Szczepaniak (Eds.)
Advances in Intelligent Web Mastering, 2007
ISBN 978-3-540-72574-9

Vol. 44. E. Corchado, J.M. Corchado, A. Abraham (Eds.)
Innovations in Hybrid Intelligent Systems, 2007
ISBN 978-3-540-74971-4

Vol. 45. M. Kurzynski, E. Puchala, M. Wozniak, A. Zolnierek (Eds.)
Computer Recognition Systems 2, 2007
ISBN 978-3-540-75174-8

Vol. 46. V.-N. Huynh, Y. Nakamori, H. Ono, J. Lawry, V. Kreinovich, H.T. Nguyen (Eds.)
Interval / Probabilistic Uncertainty and Non-classical Logics, 2008
ISBN 978-3-540-77663-5

Vol. 47. E. Pietka, J. Kawa (Eds.)
Information Technologies in Biomedicine, 2008
ISBN 978-3-540-68167-0

Vol. 48. D. Dubois, M. Asunción Lubiano, H. Prade, M. Ángeles Gil, P. Grzegorzewski, O. Hryniewicz (Eds.)
Soft Methods for Handling Variability and Imprecision, 2008
ISBN 978-3-540-85026-7

Vol. 49. J.M. Corchado, F. de Paz, M.P. Rocha, F. Fernández Riverola (Eds.)
2nd International Workshop on Practical Applications of Computational Biology and Bioinformatics (IWPACBB 2008), 2009
ISBN 978-3-540-85860-7

Vol. 50. J.M. Corchado, S. Rodriguez, J. Llinas, J.M. Molina (Eds.)
International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008), 2009
ISBN 978-3-540-85862-1

Vol. 51. J.M. Corchado, D.I. Tapia, J. Bravo (Eds.)
3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008, 2009
ISBN 978-3-540-85866-9

Vol. 52. E. Avineri, M. Köppen, K. Dahal, Y. Sunitiyoso, R. Roy (Eds.)
Applications of Soft Computing, 2009
ISBN 978-3-540-88078-3

Vol. 53. E. Corchado, R. Zunino, P. Gastaldo, Á. Herrero (Eds.)
Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems CISIS 2008, 2009
ISBN 978-3-540-88180-3

Vol. 54. B.-y. Cao, C.-y. Zhang, T.-f. Li (Eds.)
Fuzzy Information and Engineering, 2009
ISBN 978-3-540-88913-7

Vol. 55. Y. Demazeau, J. Pavón, J.M. Corchado, J. Bajo (Eds.)
7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009), 2009
ISBN 978-3-642-00486-5

Vol. 56. H. Wang, Y. Shen, T. Huang, Z. Zeng (Eds.)
The Sixth International Symposium on Neural Networks (ISNN 2009), 2009
ISBN 978-3-642-01215-0

Vol. 57. M. Kurzynski, M. Wozniak (Eds.)
Computer Recognition Systems 3, 2009
ISBN 978-3-540-93904-7

Vol. 58. J. Mehnen, M. Köppen, A. Saad, A. Tiwari (Eds.)
Applications of Soft Computing, 2009
ISBN 978-3-540-89618-0

Jörn Mehnen, Mario Köppen,
Ashraf Saad, Ashutosh Tiwari (Eds.)

Applications of Soft Computing

From Theory to Praxis

Springer

Editors

Priv.-Doz. Dr.-Ing. Jörn Mehnen
School of Applied Science (SAS)
Decision Engineering Centre
Cranfield University
Cranfield, Bedfordshire, MK43 0AL
UK
E-mail: j.mehnen@cranfield.ac.uk

Dr. Ashraf Saad
Computer Science Department
School of Computing
Armstrong Atlantic State University
Savannah, GA 31419
USA
E-mail: Ashraf.saad@armstrong.edu

Dr.-Ing. Mario Köppen
Dept. of Artificial Intelligence
Faculty of Computer Science and
Systems Engineering
Kyushu Institute of Technology Kawazu
Iizuka, Fukuoka 820-8502
Japan
E-mail: mkoeppen@
pluto.ai.kyutech.ac.jp

Dr. Ashutosh Tiwari
School of Applied Science (SAS)
Decision Engineering Centre
Cranfield University
Cranfield, Bedfordshire, MK43 0AL
UK
E-mail: a.tiwari@cranfield.ac.uk

ISBN 978-3-540-89618-0 e-ISBN 978-3-540-89619-7

DOI 10.1007/978-3-540-89619-7

Advances in Intelligent and Soft Computing ISSN 1867-5662

Library of Congress Control Number: Applied for

©2009 Springer-Verlag Berlin Heidelberg

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

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

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed in acid-free paper

5 4 3 2 1 0

springer.com

Preface

WSC 2008 Chair's Welcome Message

Dear Colleague,

The World Soft Computing (WSC) conference is an annual international online conference on applied and theoretical soft computing technology. This WSC 2008 is the thirteenth conference in this series and it has been a great success.

We received a lot of excellent paper submissions which were peer-reviewed by an international team of experts. Only 60 papers out of 111 submissions were selected for online publication. This assured a high quality standard for this online conference. The corresponding online statistics are a proof of the great world-wide interest in the WSC 2008 conference. The conference website had a total of 33,367 different human user accesses from 43 countries with around 100 visitors every day, 151 people signed up to WSC to discuss their scientific disciplines in our chat rooms and the forum. Also audio and slide presentations allowed a detailed discussion of the papers.

The submissions and discussions showed that there is a wide range of soft computing applications to date. The topics covered by the conference range from applied to theoretical aspects of fuzzy, neuro-fuzzy and rough sets over to neural networks to single and multi-objective optimisation. Contributions about particle swarm optimisation, gene expression programming, clustering, classification, support vector machines, quantum evolution and agent systems have also been received. One whole session was devoted to soft computing techniques in computer graphics, imaging, vision and signal processing.

This year WSC 2008 recognised the steadily increasing number of theoretical papers, both theoretical papers as well as practical contributions complemented very well and made WSC 2008 an open and comprehensive discussion forum.

This proceeding is a compilation of selected papers from the WSC 2008 conference now being available to the public in a printed format. We look forward to meeting you again in Cyberspace at the WSC 2009 conference.

General Chairman
Priv.-Doz. Dr.-Ing. Dipl.-Inform. Jörn Mehnen
Cranfield University, Cranfield, UK

Programme Co-Chairs
Mario Köppen, Kyushu Institute of Technology, Fukuoka, Japan
Ashraf Saad, Armstrong Atlantic State University, USA

23^{rd} *February 2009*
http://wsc-2008.softcomputing.org

Welcome Note by the World Federation on Soft Computing (WFSC) Chairman

On behalf of the World Federation on Soft Computing (WFSC) I would like to thank you for your contribution to WSC 2008! The 13th online World Conference on Soft Computing in Industrial Applications provides a unique opportunity for soft computing researchers and practitioners to publish high quality papers and discuss research issues in detail without incurring a huge cost. The conference has established itself as a truly global event on the Internet. The quality of the conference has improved over the years. The WSC 2008 conference has covered new trends in soft computing to state of the art applications. The conference has also added new features such as virtual exhibition and online presentation.

I would also like to take this opportunity to thank the organisers for a successful conference. The quality of the papers is high and the conference was very selective in accepting the papers. Also many thanks to the authors, reviewers, sponsors and publishers of WSC 2008! I believe your hard work has made the conference a true success.

Chairman of WFCS
Professor Rajkumar Roy
World Federation on Soft Computing (WFSC)
24^{th} *February 2009*

WSC 2008 Organization

Honorary Chair

Hans-Paul Schwefel	Technische Universität Dortmund, Dortmund, Germany

General Chair

Jörn Mehnen	Cranfield University, Cranfield, UK

Programme Co-Chairs

Mario Köppen	Kyushu Institute of Technology, Fukuoka, Japan
Ashraf Saad	Armstrong Atlantic State University, USA

Finance Chair

Ashutosh Tiwari	Cranfield University, Cranfield, UK

Web Coordination Chair

Lars Mehnen	Technikum Vienna, Vienna, Austria

Award Chair

Xiao-Zhi Gao	Helsinki University of Technology, Finland

Publication Chair

Keshav Dahal	University of Bradford, UK

Design Chair

Mike Goatman	Cranfield University, Cranfield, UK

International Technical Program Committee (in alphabetic order)

Ajith Abraham	University of Science and Technology, Norway
Janos Abonyi	Pannon University, Hungary
Akira Asano	Hiroshima University, Japan
Erel Avineri	University of the West of England, UK
Sudhirkumar Barai	Indian Institute of Technology Kharagpur, India
Bernard de Baets	Ghent University, Belgium
Valeriu C. Beiu	United Arab Emirates University, UAE
Alexandra Brintrup	University of Cambridge, UK
Zvi Boger	OPTIMAL - Industrial Neural Systems, Israel
Oscar Castillo	Tijuana Institute of Technology, Mexico
Sung-Bae Cho	Yonsei University, Korea
Leandro dos Santos Coelho	Pontifical Catholic University of Parana, Brazil
Carlos A. Coello Coello	CINVESTAV-IPN, Mexico
Oscar Cordon	European Centre for Soft Computing, Spain
Keshav Dahal	University of Bradford, UK
Justin Dauwels	MIT Massachusetts Institute of Technology, USA
Suash Deb	C.V. Raman College of Enineering, Bhubaneswar, India
Giuseppe Di Fatta	The University of Reading Whiteknights, UK
Matjaz Gams	Jozef Stefan Institute, Slovenia
Xiao-Zhi Gao	Helsinki University of Technology, Finland
António Gaspar-Cunha	University of Minho Campus de Azurém, Portugal
Bernard Grabot	LGP-ENIT, France
Roderich Gross	Ecole Polytechnique Fédérale de Lausanne, Switzerland

Jerzy Grzymala-Busse	University of Kansas, USA
Hani Hagras	University of Essex, United Kingdom
Ioannis Hatzilygeroudis	University of Patras, Greece
Francisco Herrera	University of Granada, Spain
Frank Hoffmann	Technical University Dortmund, Germany
Evan Hughes	Cranfield University, Shrivenham Campus, UK
Silviu Ionita	University of Pitesti, Romania
Hisao Ishibuchi	Osaka Prefecture University, Japan
Yaochu Jin	Honda Research Institute Europe, Germany
Akimoto Kamiya	Kushiro, Hokkaido, Japan
Robert Keller	Essex University, UK
Petra Kersting	Technical University Dortmund, Germany
Frank Klawonn	University of Applied Sciences Braunschweig, Germany
Andreas König	Technische Universität Kaiserslautern, Germany
Renato Krohling	Federal University of Espirito Santo, Brazil
William Langdon	Essex University, UK
Uwe Ligges	Technical University Dortmund, Germany
Luis Magdalena	European Centre for Soft Computing, Spain
Christophe Marsala	Universite Pierre et Marie Curie, France
Lars Mehnen	University of Applied Science Technikum Wien, Austria
Patricia Melin	Tijuana Institute of Technology, Mexico
Thomas Michelitsch	Technical University Dortmund, Germany
Sanaz Mostaghim	University of Karlsruhe, Germany
Mehmet Kerem Müezzinoglu	University of Louisville, USA
Zensho Nakao	University of the Ryukyus, Japan
Jae C. Oh	University Syracuse, NY, USA
Marcin Paprzycki	Polish Academy of Sciences, Poland
Petrica Pop	North University of Baia Mare, Romania
Radu-Emil Precup	University of Timisoara, Romania
Mike Preuss	Technical University Dortmund, Germany
Muhammad Sarfraz	Kuwait University, Kuwait
Dhish Saxena	Cranfield University, UK
Abhinav Saxena	RIACS - NASA ARC, USA
Giovanni Semeraro	Universita' di Bari, Italy
Yos Sunitiyoso	University of Southampton, UK
Roberto Teti	University of Naples, Italy
Ashutosh Tiwari	Cranfield University, UK
Heike Trautmann	Technical University Dortmund, Germany
Guy De Tré	Ghent University, Belgium

Eiji Uchino	Yamaguchi University, Japan
Olgierd Unold	Wroclaw University of Technology, Poland
Berend Jan van der Zwaag	University of Twente, The Netherlands
Marley Maria B.R. Vellasco	Pontificia Universidade Catolica do Rio de Janeiro, Brasil
Kostas Vergidis	Cranfield University, UK
Michael N. Vrahatis	University of Patras, Greece
Tobias Wagner	Technial University Dortmund, Germany
Matthew Wiggins	TIAX LLC, Cambridge, MA, USA

WSC 2008 Technical Sponsors

WSC 2008 was supported by:
World Federation on Soft Computing (WFSC)
IEEE Industry Application Society
IEEE UKRI Section
Elsevier

Contents

Part I: Fuzzy, Neuro-Fuzzy and Rough Sets Applications

Fuzzy Group Decision Making for Management of Oil Spill Responses 3
Renato A. Krohling, Daniel Rigo

Sensor Fusion Map Building-Based on Fuzzy Logic Using Sonar and SIFT Measurements 13
Alfredo Chávez Plascencia, Jan Dimon Bendtsen

Rough Sets in Medical Informatics Applications 23
Aboul Ella Hassanien, Ajith Abraham, James F. Peters, Gerald Schaefer

A Real Estate Management System Based on Soft Computing 31
Carlos D. Barranco, Jesús R. Campaña, Juan M. Medina

Proportional Load Balancing Using Scalable Object Grouping Based on Fuzzy Clustering 41
Romeo Mark A. Mateo, Jaewan Lee

Part II: Neural Network Applications

Multilevel Image Segmentation Using OptiMUSIG Activation Function with Fixed and Variable Thresholding: A Comparative Study 53
Sourav De, Siddhartha Bhattacharyya, Paramartha Dutta

Artificial Neural Networks Modeling to Reduce Industrial Air Pollution 63
Zvi Boger

Wavelet Neural Network as a Multivariate Processing Tool in Electronic Tongues 73
Juan Manuel Gutiérrez, Laura Moreno-Barón, Lorenzo Leija, Roberto Muñoz, Manel del Valle

Design of ANFIS Networks Using Hybrid Genetic and SVD Method for the Prediction of Coastal Wave Impacts ... 83
Ahmad Bagheri, Nader Nariman-Zadeh, Ali Jamali, Kiarash Dayjoori

A Neuro-Fuzzy Control for TCP Network Congestion 93
S. Hadi Hosseini, Mahdieh Shabanian, Babak N. Araabi

Use of Remote Sensing Technology for GIS Based Landslide Hazard Mapping 103
S. Prabu, S.S. Ramakrishnan, Hema A. Murthy, R.Vidhya

An Analysis of the Disturbance on TCP Network Congestion 115
Mahdieh Shabanian, S. Hadi Hosseini, Babak N. Araabi

Part III: Applications of Evolutionary Computations

RAM Analysis of the Press Unit in a Paper Plant Using Genetic Algorithm and Lambda-Tau Methodology 127
Komal, S.P. Sharma, Dinesh Kumar

A Novel Approach to Reduce High-Dimensional Search Spaces for the Molecular Docking Problem 139
Dimitri Kuhn, Robert Günther, Karsten Weicker

GA Inspired Heuristic for Uncapacitated Single Allocation Hub Location Problem 149
Vladimir Filipović, Jozef Kratica, Dušan Tošić, Djordje Dugošija

Evolutionary Constrained Design of Seismically Excited Buildings: Sensor Placement 159
Alireza Rowhanimanesh, Abbas Khajekaramodin, Mohammad-Reza Akbarzadeh-T.

Applying Evolution Computation Model to the Development and Transition of Virtual Community under Web2.0 171
Huang Chien-hsun, Tsai Pai-yung

Genetic Algorithms in Chemistry: Success or Failure Is in the Genes 181
Clifford W. Padgett, Ashraf Saad

Part IV: Other Soft Computing Applications

Multi-objective Expansion Planning of Electrical Distribution Networks Using Comprehensive Learning Particle Swarm Optimization 193
Sanjib Ganguly, N.C. Sahoo, D. Das

Prediction of Compressive Strength of Cement Using Gene Expression Programming 203
Priyanka Thamma, S.V. Barai

Fault-Tolerant Nearest Neighbor Classifier Based on Reconfiguration of Analog Hardware in Low Power Intelligent Sensor Systems 213
Kuncup Iswandy, Andreas König

Text Documents Classification by Associating Terms with Text Categories 223
V. Srividhya, R. Anitha

Applying Methods of Soft Computing to Space Link Quality Prediction 233
Bastian Preindl, Lars Mehnen, Frank Rattay, Jens Dalsgaard Nielsen

A Novel Multicriteria Model Applied to Cashew Chestnut Industrialization Process 243
Isabelle Tamanini, Ana Lisse Carvalho, Ana Karoline Castro, Plácido Rogério Pinheiro

Part V: Design of Fuzzy, Neuro-Fuzzy and Rough Sets Techniques

Selection of Aggregation Operators with Decision Attitudes 255
Kevin Kam Fung Yuen

A New Approach Based on Artificial Neural Networks for High Order Bivariate Fuzzy Time Series 265
Erol Egrioglu, V. Rezan Uslu, Ufuk Yolcu, M.A. Basaran, Aladag C. Hakan

A Genetic Fuzzy System with Inconsistent Rule Removal and Decision Tree Initialization 275
Pietari Pulkkinen, Hannu Koivisto

Robust Expectation Optimization Model Using the Possibility Measure for the Fuzzy Random Programming Problem 285
Takashi Hasuike, Hiroaki Ishii

Improving Mining Fuzzy Rules with Artificial Immune Systems by Uniform Population 295
Edward Mężyk, Olgierd Unold

Incremental Locally Linear Fuzzy Classifier 305
Armin Eftekhari, Mojtaba Ahmadieh Khanesar, Mohamad Forouzanfar, Mohammad Teshnehlab

On Criticality of Paths in Networks with Imprecise Durations and Generalized Precedence Relations 315
Siamak Haji Yakhchali, Seyed Hassan Ghodsypour, Seyed Mohamad Taghi Fatemi Ghomi

Part VI: Design of Evolutionary Computation Techniques

Parallel Genetic Algorithm Approach to Automated Discovery of Hierarchical Production Rules 327
K.K. Bharadwaj, Saroj

Two Hybrid Genetic Algorithms for Solving the Super-Peer Selection Problem 337
Jozef Kratica, Jelena Kojić, Dušan Tošić, Vladimir Filipović, Djordje Dugošija

A Genetic Algorithm for the Constrained Coverage Problem 347
Mansoor Davoodi, Ali Mohades, Jafar Rezaei

Using Multi-objective Evolutionary Algorithms in the Optimization of Polymer Injection Molding 357
Célio Fernandes, António J. Pontes, Júlio C. Viana, A. Gaspar-Cunha

A Multiobjective Extremal Optimization Algorithm for Efficient Mapping in Grids 367
Ivanoe De Falco, Antonio Della Cioppa, Domenico Maisto, Umberto Scafuri, Ernesto Tarantino

Interactive Incorporation of User Preferences in Multiobjective Evolutionary Algorithms 379
Johannes Krettek, Jan Braun, Frank Hoffmann, Torsten Bertram

Improvement of Quantum Evolutionary Algorithm with a Functional Sized Population 389
Tayarani Mohammad, Akbarzadeh Toutounchi Mohammad Reza

Optimal Path Planning for Controllability of Switched Linear Systems Using Multi-level Constrained GA 399
Alireza Rowhanimanesh, Ali Karimpour, Naser Pariz

Part VII: Design for Other Soft Computing Techniques

Particle Swarm Optimization for Inference Procedures in the Generalized Gamma Family Based on Censored Data 411
Mauro Campos, Renato A. Krohling, Patrick Borges

SUPER-SAPSO: A New SA-Based PSO Algorithm 423
Majid Bahrepour, Elham Mahdipour, Raman Cheloi, Mahdi Yaghoobi

Testing of Diversity Strategy and Ensemble Strategy in SVM-Based Multiagent Ensemble Learning 431
Lean Yu, Shouyang Wang, Kin Keung Lai

Probability Collectives: A Decentralized, Distributed Optimization for Multi-Agent Systems 441
Anand J. Kulkarni, K. Tai

Part VIII: Computer Graphics, Imaging, Vision and Signal Processing

Shape from Focus Based on Bilateral Filtering and Principal Component Analysis 453
Muhammad Tariq Mahmood, Asifullah Khan, Tae-Sun Choi

Detecting Hidden Information from Watermarked Signal Using Granulation Based Fitness Approximation 463
Mohsen Davarynejad, Saeed Sedghi, Majid Bahrepour, Chang Wook Ahn, Mohammad-Reza Akbarzadeht, Carlos Artemio Coello Coello

Fuzzy Approaches for Colour Image Palette Selection 473
Gerald Schaefer, Huiyu Zhou

Novel Face Recognition Approach Using Bit-Level Information and Dummy Blank Images in Feedforward Neural Network 483
David Boon Liang Bong, Kung Chuang Ting, Yin Chai Wang

ICA for Face Recognition Using Different Source Distribution Models 491
Dinesh Kumar, C.S. Rai, Shakti Kumar

Object Recognition Using Particle Swarm Optimization on Moment Descriptors 499
Muhammad Sarfraz, Ali Taleb Ali Al-Awami

Perceptual Shaping in Digital Image Watermarking Using LDPC Codes and Genetic Programming 509
Imran Usman, Asifullah Khan, Rafiullah Chamlawi, Tae-Sun Choi

Voice Conversion by Mapping the Spectral and Prosodic Features Using Support Vector Machine 519
Rabul Hussain Laskar, Fazal Ahmed Talukdar, Rajib Bhattacharjee, Saugat Das

Author Index 529

Subject Index 531

List of Contributors

Ajith Abraham
Center for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology Trondheim, Norway
ajith.abraham@ieee.org

Mojtaba Ahmadieh Khanesar
K.N. Toosi University of Technology Shariati Ave. Tehran,
16315-1355, Iran
ahmadieh@ee.kntu.ac.ir

Chang Wook Ahn
School of Information and Communication Engineering, Sungkyunkwan University,
300 Cheoncheon-dong,
Jangan-gu, Suwon 440-746, Korea
cwan@skku.edu

Mohammad-Reza Akbarzadeht
Ferdowsi University of Mashhad, Department of Electrical Engineering, Cognitive Computing Lab Azadi Square Mashhad
91775-1111, Iran
akbarzadeh@ieee.org

Ali Taleb Ali Al-Awami
Department of Electrical Engineering, King Fahd University of
Petroleum and Minerals
Dhahran, 31261 Saudi Arabia
aliawami@kfupm.edu.sa

R. Anitha
Department of M.C.A
K.S.R. College of Technology
Tiruchengode-637 215, India
aniraniraj@rediffmail.com

N. Araabi Babak
Tehran University, School of Electrical and Computer Eng.
North Amirabad
Tehran, 14395, Iran

Ahmad Bagheri
Department of Mechanical Engineering, Faculty of Engineering,
University of Guilan
Rasht, P.O. Box 3756, Iran
bagheri@guilan.ac.ir

Majid Bahrepour
Pervasive Systems Group,

Twente University
Zilverling 4013
P.O. Box 217, 7500
AE Enschede, The Netherlands
m_bahrepour@ieee.org

S.V. Barai
Indian Institute of
Technology, Kharagpur
Kharagpur 721 302 India
skbarai@civil.iitkgp.ernet.in

Carlos D. Barranco
Pablo de Olavide University
Utrera Rd. Km. 1
Sevilla, 41013, Spain
cbarranco@upo.es

M.A. Basaran
Nigde University,
Department of Mathematics
Üniveriste Kampusu
Nigde, 51000, Turkey
muratalper@yahoo.com

Torsten Bertram
Chair for Control and
Systems Engineering,
Technische Universität
Dortmund Otto-Hahn-Str. 4
44221 Dortmund
torsten.bertram@tu-dortmund.de

K.K. Bharadwaj
School of Computer and
Systems Sciences
Jawaharlal Nehru University
New Delhi 1100067, India
kbharadwaj@gmail.com

Rajib Bhattacharjee
National Institute of Technology
Silchar, Assam 788010 India
rajib13_nits@yahoo.co.in

Siddhartha Bhattacharyya
University Institute of Technology,
The University of Burdwan
Golapbag (North)
Burdwan 713104, India
sourav.de79@gmail.com

Zvi Boger
OPTIMAL Industrial Neural
Systems Ltd.
54 Rambam St.,
Beer Sheva 84243 Israel
Optimal Neural Informatics LLC
8203 Springbottom Way
Pikesville, MD, 21208, USA
optimal@peeron.com

David Boon Liang Bong
Faculty of Engineering,
Universiti Malaysia Sarawak
Kota Samarahan
94300 Kota Samarahan,
Malaysia
davidblbong@yahoo.com

Patrick Borges
Federal University of
Espírito Santo - UFES
Department of Statistics,
Av. Fernando Ferrari 514,
Campus de Goiabeiras
Vitória - ES,
CEP 29075-910, Brazil
patrick@cce.ufes.br

Jan Braun
Chair for Control and
Systems Engineering,
Technische Universität Dortmund
Otto-Hahn-Str. 4
44221 Dortmund
jan.braun@tu-dortmund.de

Jesús R. Campaña
University of Granada
Daniel Saucedo Aranda s/n
Granada, 18071, Spain
jesuscg@decsai.ugr.es

Mauro Campos
Federal University of
Espírito Santo - UFES
Department of Statistics
Av. Fernando Ferrari 514,
Campus de Goiabeiras
Vitória - ES,
CEP 29075-910, Brazil
maurocmc@cce.ufes.br

Ana Lisse Carvalho
Graduate Program in
Applied Informatics,
University of Fortaleza
Av. Washington Soares,
1321 - Bl J Sl 30
Fortaleza, 60.811-905, Brazil
ana.carvalho@serpro.gov.br

Ana Karoline Castro
Graduate Program in
Applied Informatics,
University of Fortaleza
Av. Washington Soares,
1321 - Bl J Sl 30
Fortaleza, 60.811-905, Brazil
akcastro@gmail.com

Rafiullah Chamlawi
DCIS, Pakistan Institute
of Engineering and
Applied Sciences
P.O. Nilore
45650, Islamabad, Pakistan
chamlawi@gmail.com

Raman Cheloi
Leiden University
Leiden, The Netherlands
raman@oxcart.com

Huang Chien-hsun
National Chiao Tung
University No. 1001,
Ta Hsueh Road,
Hsinchu 300
Taiwan, ROC
katwin.huang@gmail.com

Tae-Sun Choi
Gwangju Institute of
Science and Technology
261 Cheomdan-gwagiro,
Buk-gu Gwangju,
500-712, Repulic of Korea
tschoi@gist.ac.kr

Tae-Sun Choi
Gwangju Institute of
Science and Technology
261 Cheomdan-gwagiro,
Buk-gu Gwangju,
500-712, Repulic of Korea
tschoi@gist.ac.kr

Alfredo Chávez Plascencia
Aalborg University
Fredrik bajers Vej 7C
9229 Aalborg, Denmark
acp@es.aau.dk

Carlos Artemio Coello Coello
CINVESTAV-IPN, Depto.
de Computacion
Av. Instituto Politecnico
Nacional No. 2508
Col. San Pedro Zacatenco,
Mexico, D.F. 07300
ccoello@cs.cinvestav.mx

António Cunha
University of Minho
Rua Capitão Alfredo

Guimarães Guimarães,
4800-058, Portugal
agc@dep.uminho.pt

Saugat Das
National Institute of Technology
Silchar, Assam 788010
India
saugat_nits@yahoo.co.in

D. Das
Indian Institute of
Technology Kharagpur
Kharagpur, West
Bengal-721302, India
ddas@ee.iitkgp.ernet.in

Mohsen Davarynejad
Faculty of Technology,
Policy and Management,
Delft University
of Technology
Jaffalaan 5,
2628 BX, Delft,
The Netherlands
m.davarynejad@tudelft.nl

Mansoor Davoodi
Laboratory of Algorithms
and Computational Geometry
Department of Mathematics
and Computer Science
Amirkabir University
of Technology.
Hafez, Tehran, Iran
mansoorcom81@yahoo.com,
mdmonfared@aut.ac.ir

Kiarash Dayjoori
Department of Mechanical
Engineering, Faculty
of Engineering,
University of Guilan
Rasht, P.O. Box 3756, Iran
kia60day@yahoo.com

Sourav De
University Institute of
Technology, The University
of Burdwan Golapbag (North)
Burdwan 713104, India
sourav.de79@gmail.com

Ivanoe De Falco
ICAR - CNR
Via P. Castellino 111
Naples, 80131, Italy
ivanoe.defalco@na.icar.cnr.it

Antonio Della Cioppa
Natural Computation Lab,
DIIIE, University of Salerno
Via Ponte don Melillo 1
Fisciano (SA), 84084, Italy
adellacioppa@unisa.it

Jan Dimon Bendtsen
Aalborg University
Fredrik bajers Vej 7C
9229 Aalborg, Denmark
dimon@es.aau.dk

Djordje Dugošija
University of Belgrade,
Faculty of Mathematics
Studentski trg 16/IV
11 000 Belgrade, Serbia
dugosija@matf.bg.ac.yu

Paramartha Dutta
Visva-Bharati
Santiniketan 721 325,
India

Armin Eftekhari
K.N. Toosi University
of Technology Shariati Ave.
Tehran, 16315-1355, Iran
a.eftekhari@ee.kntu.ac.ir

Erol Egrioglu
Ondokuz Mayis University,
Department of Statistics
Kurupelit Samsun,
55139, Turkey
erole@omu.edu.tr

Seyed Mohamad Taghi Fatemi Ghomi
Department of Industrial Engineering, Amirkabir University of Technology
No 424, Hafez Ave
Tehran, Iran
fatemi@aut.ac.ir

Célio Fernandes
University of Minho
Rua Capitão Alfredo Guimarães Guimarães,
4800-058, Portugal
cbpf@dep.uminho.pt

Vladimir Filipović
University of Belgrade,
Faculty of Mathematics
Studentski trg 16/IV
11 000 Belgrade, Serbia
vladaf@matf.bg.ac.yu

Mohamad Forouzanfar
K.N. Toosi University of Technology
Shariati Ave. Tehran,
16315-1355, Iran
mohamad398@ee.kntu.ac.ir

Robert Günther
University of Leipzig,
Institute of Biochemistry,
Faculty of Biosciences,
Pharmacy and Psychology
Brüderstraße 34
04103 Leipzig, Germany
robguent@uni-leipzig.de

Sanjib Ganguly
Indian Institute of Technology Kharagpur Kharagpur,
West Bengal-721302,
India
sanjib@ee.iitkgp.ernet.in

Seyed Hassan Ghodsypour
Department of Industrial Engineering, Amirkabir University of Technology
No 424, Hafez Ave
Tehran, Iran
ghodsypo@aut.ac.ir

Juan Manuel Gutiérrez
Sensors & Biosensors Group,
Dept. of Chemistry.
Universitat Autònoma de Barcelona
Edifici Cn
08193 Bellaterra,
Barcelona, Spain
jmgutier27@gmail.com

Siamak Haji Yakhchali
Department of Industrial Engineering, Amirkabir University of Technology
No 424, Hafez Ave
Tehran, Iran
yakhchali@aut.ac.ir

Aladag C. Hakan
Hacettepe University,
Department of Statistics
Beytepe Kampusu
Ankara, 06800, Turkey
chaldag@gmail.com.tr

Aboul Ella Hassanien
Information Technology Department, FCI,

Cairo University
Cairo, Egypt
abo@cba.edu.kw

Takashi Hasuike
Graduate School of
Information Science
and Technology,
Osaka University
2-1 Yamadaoka, Suita
Osaka 565-0871, Uapan
thasuike@ist.osaka-u.ac.jp

A. Murthy Hema
Department of Computer
Science and Engineering
Indian Institute of
Technology Madras, Chennai
Tamil Nadu, India
hema@lantana.tenet.res.in

Frank Hoffmann
Chair for Control and
Systems Engineering,
Technische Universität
Dortmund Otto-Hahn-Str. 4
44221 Dortmund
frank.hoffmann@tu-dortmund.de

S. Hadi Hosseini
Islamic Azad University,
Science and Research branch
30 Tir Iran
sh_hosseini@itrc.ac.ir

Hiroaki Ishii
Graduate School of
Information Science and
Technology, Osaka University
2-1 Yamadaoka, Suita
Osaka 565-0871, Uapan
ishii@ist.osaka-u.ac.jp

Kuncup Iswandy
Institute of Integrated
Sensor Systems,
University of Kaiserslautern
Erwin-Schrödinger-Strasse
Kaiserslautern 67663, Germany
kuncup@eit.uni-kl.de

Ali Jamali
Department of Mechanical
Engineering, Faculty
of Engineering,
University of Guilan
Rasht, P.O. Box 3756, Iran
alijamali_mech@yahoo.com

Andreas König
Institute of Integrated
Sensor Systems,
University of Kaiserslautern
Erwin-Schroedinger-Strasse
Kaiserslautern 67663, Germany
koenig@eit.uni-kl.de

Ali Karimpour
Ferdowsi University of
Mashhad, Department
of Electrical Engineering
Azadi Square
Mashhad, Iran
karimpor@um.ac.ir

Abbas Khajekaramodin
Ferdowsi University
of Mashhad,
Department of
Civil Engineering
Azadi Square
Mashhad, Iran
karamodin@gmail.com

Asifullah Khan
DCIS, Pakistan Institute
of Engineering and
Applied Sciences PO Nilore
45650, Islamabad, Pakistan
asif@pieas.edu.pk

Asifullah Khan
Gwangju Institute of

Science and Technology
261 Cheomdan-gwagiro,
Buk-gu Gwangju,
500-712, Repulic of Korea
asifullah@gist.ac.kr

Hannu Koivisto
Tampere University of Technology
P.O. Box 692
FI-33101 Tampere, Finland
hannu.koivisto@tut.fi

Jelena Kojić
University of Belgrade,
Faculty of Mathematics
Studentski trg 16/IV
11 000 Belgrade, Serbia
k_jelena@yubc.net

Indian Institute of Technology Roorkee (IITR) Komal
Department of Mathematics
Roorkee (Uttarakhand),
247667, India
karyadma.iitr@gmail.com

Jozef Kratica
Mathematical Institute,
Serbian Academy of
Sciences and Arts
Kneza Mihajla 36/III
11 001 Belgrade, Serbia
jkratica@mi.sanu.ac.rs

Johannes Krettek
Chair for Control and
Systems Engineering,
Technische Universität
Dortmund Otto-Hahn-Str. 4
44221 Dortmund
johannes.krettek@tu-dortmund.de

Renato Krohling
Federal University of
Espírito Santo - UFES
Department of Informatics,
PPGI Av. Fernando
Ferrari s/n. CT VII
Vitória - ES,
CEP 29060-970, Brazil
krohling.renato@gmail.com

Dimitri Kuhn
HTWK Leipzig University
of Applied Science,
Department of Computer
Science, Mathematics
and Natural Sciences
Karl-Liebknecht-Straße 132
04277 Leipzig, Germany
dkuhn@imn.htwk-leipzig.de

Anand J. Kulkarni
Nanyang Technological University
50 Nanyang Avenue
Singapore 639798, Singapore
kulk0003@ntu.edu.sg

Shakti Kumar
Institute of Science
and Technology Klawad
District Yamuna Nagar
Haryana, India
shakti@istk.org

Dinesh Kumar
Department of Computer
Science and Engineering
Guru Jambheshwar
University of Science
and Technology Hisar,
Haryana, India - 125001
dinesh_chutani@yahoo.com

Dinesh Kumar
Indian Institute of
Technology Roorkee (IITR)
Department of Mechanical
and Industrial Engineering
Roorkee (Uttarakhand),

247667, India
dinesfme@iitr.ernet.in

Kin Keung Lai
City University of Hong Kong
83 Tat Chee Avenue
Kowloon, Hong Kong, China
mskklai@cityu.edu.hk

Rabul Hussain Laskar
National Institute of
Technology Silchar,
Assam 788010
India
laskar_r@nits.ac.in

Jaewan Lee
School of Electronic
and Information
Engineering, Kunsan
National University
San 68 Miryong-dong
Kunsan city, Chonbuk
573-701, South Korea
jwlee@kunsan.ac.kr

Lorenzo Leija
Bioelectronics Section,
Department of Electrical
Engineering, CINVESTAV
Av. IPN 2508
07360, México City,
México
lleija@cinvestav.mx

Edward Mężyk
Institute of Computer
Engineering, Control
and Robotics, Wroclaw
University of Technology
Wyb. Wyspianskiego 27,
50-370 Wroclaw, Poland
edward.mezyk@pwr.wroc.pl

Elham Mahdipour
Khavaran University
Mashhad, Iran
elham_mahdipour@yahoo.com

Muhammad Tariq Mahmood
Gwangju Institute of Science
and Technology
261 Cheomdan-gwagiro,
Buk-gu Gwangju,
500-712,
Repulic of Korea
tariq@gist.ac.kr

Domenico Maisto
ICAR - CNR
Via P. Castellino 111
Naples, 80131, Italy
domenico.maisto@na.icar.cnr.it

Romeo Mark Mateo
School of Electronic
and Information Engineering,
Kunsan National University
San 68 Miryong-dong
Kunsan city,
Chonbuk 573-701,
South Korea
rmmateo@kunsan.ac.kr

Juan M. Medina
University of Granada
Daniel Saucedo Aranda s/n
Granada, 18071, Spain
medina@decsai.ugr.es

Lars Mehnen
Vienna Technical
University/Institute of
Analysis and Scientific
Computing Karlsplatz 13
Vienna, 1030, Austria
bastian@preindl.net

Ali Mohades
Laboratory of Algorithms
and Computational Geometry
Department of Mathematics

and Computer Science
Amirkabir University of Technology.
Hafez, Tehran, Iran
mohades@aut.ac.ir

Tayarani Mohammad
Azad University of Mashhad
Department of Computer Science
Mashhad, Iran
tayarani@ieee.org

Akbarzadeh Toutounchi Mohammad Reza
Ferdowsi University of
Mashhad Department of
Electrical Engineering
akbarzadeh@ieee.org

Laura Moreno-Barón
Sensors & Biosensors Group,
Dept. of Chemistry.
Universitat Autònoma
de Barcelona
Edifici Cn
08193 Bellaterra,
Barcelona, Spain
lauramorenob@gmail.com

Roberto Muñoz
Bioelectronics Section,
Department of Electrical
Engineering, CINVESTAV
Av. IPN 2508
07360, México
City, México
rmunoz@cinvestav.mx

Nader Nariman-zadeh
Department of Mechanical
Engineering, Faculty
of Engineering,
University of Guilan
Rasht, P.O. Box 3756, Iran
nnzadeh@guilan.ac.ir

Jens D. Nielsen
Aalborg University/Department
of Electronic Systems
Fredrik Bajers Vej 7C
9220 Aalborg, Denmark
bastian@preindl.net

Clifford Padgett
Armstrong Atlantic State University
11935 Abercorn Street
Savannah, GA 31419-1997 USA
clifford.padgett@armtrong.edu

Tsai Pai-yung
National Chiao Tung
University No. 1001,
Ta Hsueh Road,
Hsinchu 300 Taiwan, ROC
halohowau@gmail.com

Naser Pariz
Ferdowsi University
of Mashhad,
Department of Electrical
Engineering Azadi Square
Mashhad, Iran
n-pariz@um.ac.ir

James F. Peters
Computational Intelligence
Laboratory, Department
of Electrical and
Computer Engineering,
University of Manitoba
Winnipeg, Canada
jfpeters@ee.umanitoba.ca

Plácido Rogério Pinheiro
Graduate Program in
Applied Informatics,
University of Fortaleza
Av. Washington Soares,
1321 - Bl J Sl 30
Fortaleza, 60.811-905,
Brazil
placido@unifor.br

António Pontes
University of Minho
Rua Capitão Alfredo Guimarães

Guimarães, 4800-058, Portugal
pontes@dep.uminho.pt

S. Prabu
Institute of Remote Sensing
College of Engineering Guindy
Anna University, Chennai
Tamil Nadu, India
sevu_prabu@yahoo.co.in

Bastian Preindl
Vienna Technical
University/Institute of
Analysis and Scientific
Computing Karlsplatz 13
Vienna, 1030, Austria
bastian@preindl.net

Pietari Pulkkinen
Tampere University of
Technology P.O. Box 692
FI-33101 Tampere, Finland
pietari.pulkkinen@tut.fi

C.S. Rai
University School of
Information Technology
Guru Gobind Singh
Indraprastha University
Kashmere Gate,
Delhi, India - 110403
csrai_ipu@yahoo.com

S.S Ramakrishnan
Institute of Remote Sensing
College of Engineering Guindy
Anna University, Chennai
Tamil Nadu, India
ssramki@annauniv.edu

Frank Rattay
Vienna Technical
University/Institute of
Analysis and Scientific
Computing Karlsplatz 13
Vienna, 1030, Austria
bastian@preindl.net

Jafar Rezaei
Faculty of Technology,
Policy and Management
Delft University of
Technology P.O. Box
5015, 2600
GA Delft, The Netherlands
j.rezaei@tudelft.nl

Daniel Rigo
Federal University
of Espírito
Santo - UFES
Department of
Environmental Engineering,
PPGEA Av. Fernando
Ferrari 514, Predio CT
Vitória - ES,
CEP 29075-910, Brazil
rigo@npd.ufes.br

Alireza Rowhanimanesh
Ferdowsi University of
Mashhad, Department of
Electrical Engineering,
Cognitive Computing Lab
Azadi Square
Mashhad, Iran
rowhanimanesh@gmail.com

Ashraf Saad
Armstrong Atlantic
State University
11935 Abercorn Street
Savannah, GA 31419-1997 USA
ashraf.saad@armstrong.edu

N.C. Sahoo
Indian Institute of Technology
Kharagpur Kharagpur,
West Bengal-721302, India
ncsahoo@ee.iitkgp.ernet.in

Muhammad Sarfraz
Department of Information Science, Kuwait University, Adailiyah Campus P.O. Box 5969, Safat 13060, Kuwait
prof.m.sarfraz@gmail.com

Saroj Saroj
Department of Computer Science and Engineering Guru Jambheshwar University of Science and Technology Hisar 125001, Haryana, India
ratnoo.saroj@gmail.com

Umberto Scafuri
ICAR - CNR
Via P. Castellino 111
Naples, 80131, Italy
umberto.scafuri@na.icar.cnr.it

Gerald Schaefer
School of Engineering and Applied Science, Aston University
Birmingham, UK
g.schaefer@aston.ac.uk

Saeed Sedghi
Department of Computer Science, University of Twente, PO Box 217, 7500 AE Enschede, The Netherlands
s.sedghi@utwente.nl

Mahdieh Shabanian
Islamic Azad University, Science and Research branch
30 Tir Iran
m_shabanian@itrc.ac.ir

S.P. Sharma
Indian Institute of Technology Roorkee (IITR)
Department of Mathematics
Roorkee (Uttarakhand), 247667, India
sspprfma@iitr.ernet.in

V. Srividhya
Department of Computer Science Avinashilingam University for Women
Coimbatore 641 043, India
vidhyavasu@gmail.com

Kang Tai
Nanyang Technological University
50 Nanyang Avenue
Singapore 639798, Singapore
mktai@ntu.edu.sg

Fazal Ahmed Talukdar
National Institute of Technology Silchar, Assam 788010 India
fazal@nits.ac.in

Isabelle Tamanini
Graduate Program in Applied Informatics, University of Fortaleza
Av. Washington Soares, 1321 - Bl J Sl 30
Fortaleza, 60.811-905, Brazil
isabelle.tamanini@gmail.com

Ernesto Tarantino
ICAR - CNR
Via P. Castellino 111
Naples, 80131, Italy
ernesto.tarantino@na.icar.cnr.it

Mohammad Teshnehlab
K.N. Toosi University of Technology Shariati Ave.
Tehran, 16315-1355, Iran
teshnehlab@eetd.kntu.ac.ir

Priyanka Thamma
Indian Institute of

Technology, Kharagpur
Kharagpur 721 302
India
priyanka.t29@gmail.com

Kung Chuang Ting
Faculty of Engineering,
Universiti Malaysia Sarawak
Kota Samarahan 94300
Kota Samarahan, Malaysia
tkchuang2000@hotmail.com

Dušan Tošić
University of Belgrade,
Faculty of Mathematics
Studentski trg 16/IV
11 000 Belgrade, Serbia
tdusan@mi.sanu.ac.rs

Olgierd Unold
Institute of Computer
Engineering, Control
and Robotics, Wroclaw
University of Technology
Wyb. Wyspianskiego 27,
50-370 Wroclaw, Poland
olgierd.unold@pwr.wroc.pl

V. Rezan Uslu
Ondokuz Mayis University,
Department of Statistics
Kurupelit
Samsun, 55139, Turkey
erole@omu.edu.tr

Imran Usman
DCIS,Pakistan Institute
of Engineering and
Applied Sciences
PO Nilore
45650, Islamabad, Pakistan
imran.usman@gmail.com

Júlio Viana
University of Minho
Rua Capitão Alfredo
Guimarães Guimarães,
4800-058, Portugal
jcv@dep.uminho.pt

R. Vidhya
Institute of Remote Sensing
College of Engineering Guindy
Anna University, Chennai
Tamil Nadu, India
rvidhya@annauniv.edu

Shouyang Wang
Chinese Academy of Sciences
55 Zhongguancun East Road
Haidian District,
Beijing 100190,
China
sywang@amss.ac.cn

Yin Chai Wang
Faculty of Computer
Science and Information
Technology, Universiti
Malaysia Sarawak
Kota Samarahan
94300 Kota Samarahan,
Malaysia

Karsten Weicker
HTWK Leipzig University
of Applied Science,
Department of Computer
Science, Mathematics
and Natural Sciences
Karl-Liebknecht-Straße
132 04277 Leipzig, Germany
weicker@imn.htwk-leipzig.de

Mahdi Yaghoobi
Islamic Azad University
of Mashhad Mashhad, Iran
yaghobi@mshdiau.ac.ir

Ufuk Yolcu
Ondokuz Mayis University,
Department of Statistics
Kurupelit
Samsun, 55139, Turkey
erole@omu.edu.tr

Lean Yu
Chinese Academy of Sciences
55 Zhongguancun East Road
Haidian District,
Beijing 100190, China
yulean@amss.ac.cn

Kevin Kam Fung Yuen
The Hong Kong
Polytechnic University,
Hung Hom, Kowloon,
Hong Kong, China
kevinkf.yuen@gmail.com

Huiyu Zhou
School of Engineering
and Design, Brunel
University Uxbridge,UK
huiyu.zhou@brunel.ac.uk

Manel del Valle
Sensors & Biosensors Group,
Dept. of Chemistry.
Universitat Autònoma de
Barcelona Edifici Cn
08193 Bellaterra,
Barcelona, Spain
manel.delvalle@uab.es

Fuzzy Group Decision Making for Management of Oil Spill Responses

Renato A. Krohling and Daniel Rigo

Abstract. The selection of combat strategy to oil spill when multi-criteria and multi-person are involved in the decision process is not an easy task. In case of oil spill, urgent decisions must be made so that the available options of responses are activated in such a way that the environmental, social and economic impacts are minimized. In this context, the decision agents involved in the decision process are the environmental agency, a non-governmental organization (NGO), and a company that get in conflict during the decision process because each one defends its own interests. So, a consensus to reach the best viable solution is desirable. The advantages and disadvantages of different types of combat strategy should be weighted, taking into account the preferences and the different point of view of the decision agents. In this context, the process to form a consensus and to elaborate the response strategies necessarily involves a process of decision making with multi-objectives and multi-person (decision agents) so that the importance of social, economic and environmental factors is considered. In this work, the fuzzy evaluation method is applied in order to automatically find the best combat response. The method is applied to evaluate combat response to oil spill in the south coast of the Espirito Santo state, Brazil. Simulation results show the viability of the method.

1 Introduction

The increase in the oil activities in the coast of Espirito Santo, Brazil [1] either in the offshore production or in the transport of oil and its derivate products also increases

Renato A. Krohling
Department of Informatics, PPGI. Federal University of Espírito Santo - UFES. Av. Fernando Ferrari s/n - CT VII, Goiabeiras, CEP 29060-970, Vitória - ES, Brazil
e-mail: krohling.renato@gmail.com

Daniel Rigo
Department of Environmental Engineering. Federal University of Espírito Santo - UFES. Av. Fernando Ferrari 514, Goiabeiras, CEP 29075-910, Vitória - ES, Brazil
e-mail: rigo@npd.ufes.br

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 3–12.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

the risk of oil spills in the sea. An eventual accident with crude oil spill can cause negative environmental impacts and serious social and economic consequences for the coastal region affected. Therefore, the arrival of oil spots to the coast would affect fishing, tourism and leisure activities in beaches and would especially cause environmental damages to the local ecosystem affecting reproduction of sea species, among others.

In order to act in emergency situations, it is necessary to have a contingency plan aiming to minimize the damages caused by oil spills. In Brazil, the development of these plans has been a duty of companies involved in the exploration and transport of oil activities. The National Environmental Council, CONAMA [2] elaborated legislations and guidelines that must be followed by the companies. The design of near-optimal combat responses is difficult to be achieved because it depends on factors, such as amount and type of spilled oil, the local of the spill, climatic and oceanic conditions, among others. The problem is especially difficult to be treated due to the dynamic nature of the marine environment. Trajectories prediction of oil spot is an essential element to be used in the management of oil spills in a coastal zone. Computational models for simulation based on Lagrangian and Eulerian approaches have been used with promising results [3-9]. In case of oil spill, the interests of the environmental agencies, non-governmental organizations and of oil companies involved in the accident get inevitably in conflict in the decision process of choosing the best combat strategy; each one representing its own interests. In such situations, a consensus to reach the best viable solution is of great importance. The advantages and disadvantages of different types of combat responses should be weighted. On the one hand, one has equipment costs and maintenance, and on the other hand one wants the smallest environmental impact on the coastal areas affected by the accident. In this context, the process to reach a consensus and to elaborate response strategies necessarily involves a process of decision making with multiples objectives that take into account the preferences of the decision agents [10], so that the importance of social, economic and environmental factors is considered.

In emergency cases, the effective response capacity to oil spill in the open sea reveals a critical issue in the domain of integrated management of coastal regions. On the one hand, the goal is to remove the highest possible amount of spilled oil in the surface of the sea in order to minimize the environmental impact on the coast; and one the other hand the goal is also minimize the investments on equipments and facilities and its maintenance and cleaning costs. In this case study, various combat strategies for a hypothetical oil spill in the field of Jubarte, ES, Brazil, are simulated. The goal consists in the elaboration of a decision matrix. In this way, different alternative solutions will be performed, and in turn, selected according to environmental impacts, and social and economic aspects. Since the interests of the decision agents are involved and are directly or indirectly affected by the decisions made, a methodology to take into account the linguistic preferences of the decision makers based on fuzzy logic is presented.

The goal of this work is to develop a tool for decision support to aid the involved decision makers in the contingency plan, which consists of combat strategies to oil spill for the south coast of the ES. The system should be able to suggest good

combat alternatives based on preferences of the decision agents. Such preferences expressed through linguistic terms are described by means of fuzzy logic. The linguistic knowledge of the group is aggregated providing good combat alternatives to oil spills. The remainder of this paper is organized as follows: Section 2 shortly describes decision making methods; the case study is presented in section 3. In section 4, the approaches using the fuzzy evaluation method are described. Simulation results are shown in section 5 and conclusions with directions for future works are given in section 6.

2 Fuzzy Decision Making Methods

Fuzzy models [11] provide a way to incorporate information to an uncertainty model when statistical information is not available, or when it is necessary to deal with qualitative descriptions provided by specialists in form of declarations regarding impact of alternatives. In this work, it is enough to only consider that it is possible to evaluate the impact of an alternative through a fuzzy set. Thus, each alternative has a degree of membership in the fuzzy decision. So, it is possible an interactive intervention by the decision agents, who can modify the definitions of goals and choose acceptable levels during the decision process. In the construction of a model with multiple criteria [12] one has in mind the acceptable goals and thresholds. Fuzzy logic tries to capture the complexity of the evaluation process through linguistic declarations.

The performance of the alternatives in terms of criteria contributes to form the matrix of inputs of a fuzzy model. Additionally, the preferences of the decision agents for each pre-specified criterion are required. The values of criteria may be disclosed in a qualitative or quantitative way. In many cases, it is not realistic to expect that participants of a decision process, who have no technical background, provide a numerical value for a specified criterion. Therefore, the participants of the decision process are guided to select an importance level and their preferences are directly integrated in the fuzzy decision model [13].

By means of numerical simulation, the consequences of the use of different combat strategies in function of the specified criterion can be evaluated. Generally, it is possible to characterize scenarios through linguistic descriptions. Thus, in an impact model it is always necessary to transform this approach by providing means to evaluate the consequences of a decision for each possible scenario. In this work, the impact is described by the environmental damage [14-15].

3 Case Study

The oil field of Jubarte is one of the largest Brazilian oil reservoirs supplying heavy oil (17^{o} API); situated in the north of the geologic basin of Campos, state of Rio de Janeiro, in deep waters (about 1,300m deep) in the south coast of the state of Espirito Santo. It is situated approximately 80 km off the coast to the Pontal de

Ubú, as shown in Fig. 1 [16]. The geodesic coordinates from which the simulations of combat responses to a hypothetical accident with heavy oil have been made are: 21°15'33.2" S and 40°01'02" W. For investigation purposes, the volume of spilled oil was fixed in 15,000 m^3, which represents the scenario of the highest amount of oil reaching the coast (worst scenario). The environmental conditions and other parameters of the hydro-dynamical model have been kept the same as described in Environmental Impact Assessment (EIA) of Jubarte [17]. The main parameters used in the modeling of combat strategies to the hypothetical oil spill in Jubarte oil field are found in [16].



Fig. 1 Location of the Jubarte oil field

Response strategies to oil spill have been studied by simulating trajectories of oil spot based on information of the type of oil, location of the spill point, spilled volume, meteorological and oceanic conditions [16]. This kind of study allows simulating different scenarios (including non-response) and some configurations of combat strategies using different types of equipments (containment boats, collect boats, mechanical and chemical dispersion, etc.). In order to elaborate a contingency plan, the information provided by the prediction of spot trajectories is of fundamental importance. In this study, the scenarios have been simulated using the computational package OILMAP [18]. After preliminary evaluations, some alternatives characterized for certain number of contention formations (scenarios) will be pre-selected. Here alternatives and solutions will always be synonyms of possible decisions. For this case study, 10 alternatives have been defined. The set of alternatives is evaluated according to each of the criteria, which may be considered as representative for many coastal regions. As examples of these criteria, one can use the amount of oil that reaches the coast, the oil collected by the combat formations, the cleaning costs, among others. They reflect the interests with special emphasis on the pollution due to the spilled oil. Basically, the process of decision making is composed of decision agents, alternatives and criteria. In this study, the alternatives represent the number of formations (equipments); the criteria are oil at the coast, and oil intercepted; and

the decision agents are represented by an environmental agency, an NGO, and an oil company). Detailed Information on the formation of the alternatives may be found in [16]. Table 1 presents the result of the simulations for several combat alternatives based on containment and collection associated with the combined dispersion (mechanical and chemical) of the spilled oil. Although the criteria adopted in this work is the amount of oil at the coast (OC) and the amount of oil intercepted (OI) representing the amount of oil collected plus the amount of oil dispersed, other criteria may be used.

Table 1 Combat alternatives and criteria

Alternatives	Oil at the cost (OC) in m^3	Oil Intercepted (OI) (collected + dispersed) in m^3
Alt. 1	8.627	5.223
Alt. 2	9.838	4.023
Alt. 3	10.374	3.495
Alt. 4	8.200	5.659
Alt. 5	5.854	7.989
Alt. 6	8.108	5.790
Alt. 7	6.845	7.083
Alt. 8	5.738	8.238
Alt. 9	5.858	8.189
Alt. 10	6.269	7.808

4 Fuzzy Evaluation Method

The description of the Fuzzy Evaluation Method (FEM) [19] presented here is based on the approach proposed in [14]. Basically, the first stage consists of a performance matrix obtained via simulations of the combat strategies (alternatives) that contain uncertainties and that are fuzzyfied. In the second stage, the participants of the decision process inform their preferences in linguistic terms, which are defuzzified and integrated to the decision model. This way, the global impact of each alternative is calculated as shown in Fig. 2.

The first step is to establish the number of fuzzy sets for each criterion. Usually, a number between 3 and 7 fuzzy sets is enough. Five fuzzy sets were defined to describe the level of impact for the criterion oil at the coast (OC): very low (VL), low (L) medium (M), high (H), and very high (VH). So, the number of fuzzy sets for the criterion μ_i is $\{\mathrm{VL}_i\ \mathrm{L}_i\ \mathrm{M}_i\ \mathrm{H}_i\ \mathrm{VH}_i\}$. The second step is to choose the membership functions for each fuzzy set for the criterion i, so that the numerical values of the performance matrix are fuzzified. Triangular, trapezoidal, or Gaussian membership functions may be used. Let x_i^n assign the value of the performance for the alternative n in terms of the criterion i, then μ_{ij}^n indicates the membership degree regarding the j-th fuzzy set of the i-th criterion.

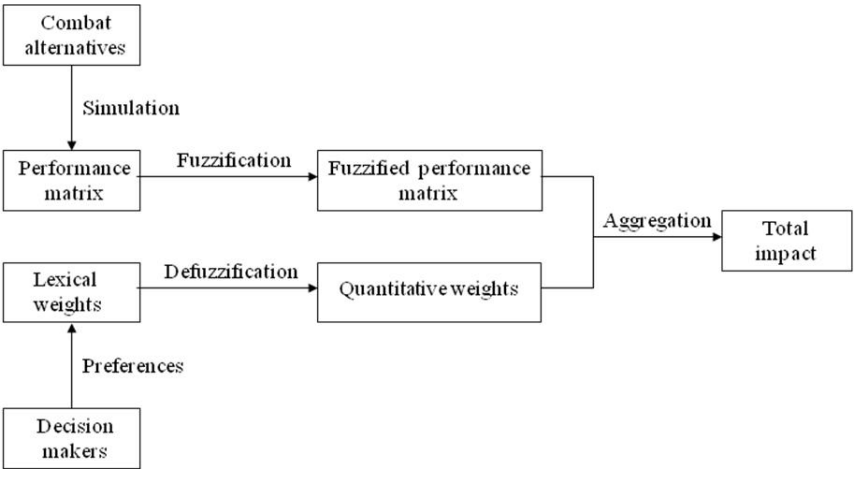


Fig. 2 Fuzzy Evaluation Method for group decision making

The next step consists in quantifying the environmental damage in terms of the combat strategy. An easy way used in [14] is to introduce levels of environmental damages equal to zero (no impact) until to one (maximum impact) equally spaced expressed by the vector $[\delta_1,\delta_2 ,...,\delta_{11}] = [0, 0.1,...,0.9, 1.0]$. In the following, the Fuzzy Evaluation Method consisting of two stages is described.

4.1 First Order Fuzzy Evaluation Method

The assignment matrix of the fuzzy degree of the 1st order represents lexical degrees associated to the levels of environmental damages (vectord δ). For the criterion OC the assignment matrix is shown in Table 2. Although representing specialist knowledge of the problem, the coefficients are of empirical nature and may be modified in an interactive way according to the application. This way, the combination of the assignment matrix of the fuzzy degree of 1st order (R_i) with the damage fuzzy vector (A_i^n) results in the set of the 1st order evaluation that can be calculated for the alternative n in terms of criterion i by

$$B_i^n = A_i^n \cdot R_i^n \tag{1}$$

4.2 Second Order Fuzzy Evaluation Method

In the process of decision making for the management of oil spill responses, it is evident that for each criterion as OC, and OI, the perspective of the decision agents (environmental agency, NGO, and oil company) is not given the same importance. Therefore, following the approach developed in [14] a weight vector $W^{s\,n}$ is

Table 2 Damage levels and fuzzyfication [14]

Fuzzy sets	Damage levels (11)										
VL	1	0.8	0.6	0.4	0	0	0	0	0	0	0
L	0.6	0.8	1	0.8	0.6	0.4	0	0	0	0	0
M	0	0	0.4	0.6	0.8	1	0.8	0.6	0.4	0	0.6
H	0	0	0	0	0	0.4	0.6	0.8	1	0.8	1
VH	0	0	0	0	0	0	0	0.4	0.6	0.8	1

introduced to denote the weight for criterion n regarding the opinion of the decision agent s. Three levels of importance are assigned for each criterion: *very important*, *moderate* and *unimportant*. In this work, the number of decision agents is three (the environmental agency, NGO, and oil company). They express their preferences according to Table 3. Finally, to transform lexical information into quantitative numerical information, the defuzzyfication method used is the weighted average. Thus, by multiplying the vector of weight $W^{s\,n}$ by the 1st order evaluation set B_i, results in the 2nd order evaluation set $K^{s\,n}$, that can be calculated for the alternative n according to the opinion of the decision agent s as

$$K^{s\,n} = W^{s\,n} \cdot B^n = k_1^{s\,n} \;\; k_2^{s\,n} \qquad k_{11}^{s\,n} \tag{2}$$

Table 3 Opinions of the decision agents in form of weights (adapted from [14])

Decision makers	Criterium OC	Criterium OI
Agent 1: Environmental agency	*moderate*	*moderate*
Agent 2: Oil company	*moderate*	*very important*
Agent 3: NGO	*very important*	*unimportant*

For the labels *very important*, *moderate* and *unimportant* are assigned the weights 0.95, 0.5, 0.05, respectively.

The global impact for each alternative n taking into account the opinion of the decision agent s is calculated by

$$K^{s\,n} = \frac{\sum_{p=1}^{11} K_p^{s\,n} \cdot \delta_p}{\sum_{p=1}^{11} \delta_p} \cdot \tag{3}$$

After calculating the global impact it is possible to rank the alternatives; those that present smaller global impact should be preferred. In the following, results of a hypothetical oil spill described in section 3 as case study are presented in order to illustrate the method.

5 Results

The data shown in Table 1, provided by means of numerical simulations of the oil spot trajectories for 10 alternatives, serve as input for the decision process. For the two criteria, oil at the coast (OC) and oil intercepted (OI) fuzzy sets with Gaussian membership functions were used, as shown in Fig. 3.

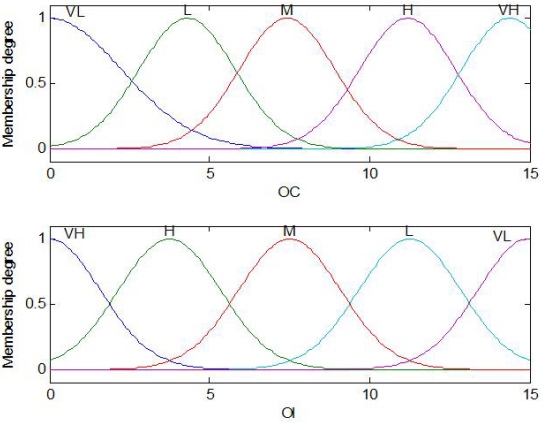


Fig. 3 Membership functions for the linguistic variable oil at the coast (OC) and for the linguistic variable oil intercepted (OI), each with 5 fuzzy sets

This type of membership function is very used due to its overlapping characteristics. The fuzzy sets VL, L, M, H, VH stands for "Very Low", "Low", "Medium", "High" and "Very High" impact, respectively. The three participants represented by an environmental agency, an oil company, and a NGO express their preferences as given in Table 3. The application of the fuzzy evaluation method allows calculating the total impact for each alternative n taking into account the preference of the decision agent s, as shown in Table 4. After ranking the alternatives for each decision agent one gets the classification shown in Table 5. The solution is given by alternatives of lesser total impact.

Table 4 Total impact of 10 alternatives for each of the 3 decision agents

Decision makers	Alt.1	Alt.2	Alt.3	Alt.4	Alt.5	Alt.6	Alt.7	Alt.8	Alt.9	Alt.10
Agent 1	0.67	0.66	0.63	0.66	0.48	0.66	0.60	046	0.47	0.53
Agent 2	1.00	0.96	0.89	1.00	0.74	0.99	0.90	0.70	0.71	0.80
Agent 3	0.60	0.65	0.69	0.60	0.41	0.61	0.55	0.40	0.41	0.47

As we can observe in Table 5, the best option for agent 1 (environmental agency), for agent 2 (the oil company) and for agent 3 (the NGO representing ecologists) is Alternative 8 that represents the smaller global impact. By changing the weights

Table 5 Ranking of the 10 alternatives taking into account the preferences of the 3 decision agents

Decision makers	Alternatives								
Agent 1	Alt.8	Alt.9	Alt.5	Alt.10	Alt.7	Alt.3	Alt.2	Alt.6	Alt.4
	Alt.1								
Agent 2	Alt.8	Alt.9	Alt.5	Alt.10	Alt.3	Alt.7	Alt.2	Alt.6	Alt.4
	Alt.1								
Agent 3	Alt.8	Alt.9	Alt.5	Alt.10	Alt.7	Alt.1	Alt.4	Alt.6	Alt.2
	Alt.3								

(opinions of the decision makers), automatically one alters the preferences and the calculation of the total impact of the alternatives, which, in turn, may modify the ranking of the alternatives.

6 Conclusions

In this work, a computational method for supporting decision making applied to management of oil spill response is presented. The method allows the participants of the decision process to express their preferences in linguistic form. In this manner it is possible to consider information of different groups of participants and to combine environmental, and others criteria through the use of fuzzy logic. An application to a case study involving a hypothetical accident in the oil field of Jubarte, situated in the south coast of Espirito Santo State, Brazil, shows the suitability of the method that serves as aid in the identification of good combat strategies (alternatives). For this case study, ten combat alternatives, two decision criteria, and three participants representing groups of decision makers have been used. However, the method can be applied for any number of participants, alternatives and criteria. Methods for ranking fuzzy numbers are being investigated and will be reported in the future.

Acknowledgements. R. A. Krohling thanks the financial support of the FAPES/MCT/CNPq (DCR grant 37286374/2007) for funding this research project in form of a fellowship. The authors acknowledge J. P. Ferreira, Petrobras, UN-ES, Vitória, Brazil, for his simulation studies of the hypothetical oil spill.

References

1. ANP: Agência Nacional do Petróleo. Brasília, Brasil. Anuário Estatístico 2004. Distribuição Percentual das Reservas Provadas de Petróleo, segundo Unidades da Federação (2004) (in Portuguese)
2. Conselho Nacional do Meio Ambiente, Brasília, Brasil. Resolução N^0 293, de 12 de dezembro de 2001. – ANEXO III Critérios para o Dimensionamento da Capacidade Mínima de Resposta, p. 14, DOU 29/04/2002 (in Portuguese)

3. Mackay, D., Paterson, S., Trudel, K.: A Mathematical Model of Oil Spill Behavior. Department of Chemical Engineering, University of Toronto, Canada (1980a)
4. Mackay, D., Paterson, S., Trudel, K.: Oil Spill Processes and Models. Report EE-8, Environmental Protection Service, Canada (1980b)
5. Mackay, D., Shui, W., Houssain, K., Stiver, W., McCurdy, D., Paterson, S.: Development and Calibration of an Oil Spill Behavior Model. Report No. CG-D027-83, US Coast Guard Research and Development Center, Groton, CT (1982)
6. McCay, D.F., Rowe, J.J., Whittier, N., Sankaranarayanan, S., Etkin, D.S.: Estimation of Potential Impacts and Natural Resource Damages of Oil. Journal of Hazardous Materials 107, 11–25 (2004)
7. Reed, M., Aamo, O.M., Daling, P.S.: Quantitative Analysis of Alternate Oil Spill Response Strategies using OSCAR. Spill Science & Technology Bulletin 2, 67–74 (1995)
8. Reed, M., Daling, P.S., Lewis, A., Ditlevsen, M.K., Brors, B., Clark, J., Aurand, D.: Modelling of Dispersant Application to Oil Spills in Shallow Coastal Waters. Environmental Modelling and Software 19, 681–690 (2004)
9. Lehr, W., Jones, R., Evans, M., Simeck-Beatty, D., Overstreet, R.: Revisions of the ADIOS Oil Spill Model. Environmental Modelling and Software 17, 191–199 (2002)
10. Yu, Y., Wang, B., Wang, G., Li, W.: Multi-person Multiobjective Fuzzy Decision Making Model for Reservoir Flood Control Operation. Water Resources Management 18, 111–124 (2004)
11. Kaufmann, A., Gupta, M.M.: Fuzzy Mathematical Models in Engineering and Management Science. North-Holland, Amsterdam (1988)
12. Zimmermann, H.-J.: Fuzzy Sets, Decision Making, and Expert Systems. Kluwer, Boston (1987)
13. Matos, M.A.: Eliciting and Aggregating Preferences with Fuzzy Inference Systems. In: Antunes, C.H., Figueira, J., Clýmaco, J. (eds.) Multiple Criteria Decision Making, pp. 213–227. CCDRC/FEUC, Coimbra (2004)
14. Liu, X., Wirtz, K.W.: Consensus Oriented Fuzzified Decision Support for Oil Spill Contingency Management. Journal of Hazardous Materials 134, 27–35 (2006)
15. Wirtz, K.W., Baumberger, N., Adam, S., Liu, X.: Oil Spill Impact Minimization under Uncertainty: Evaluating Contingency Simulations of the Prestige Accident. Ecological Economics 61, 417–428 (2007)
16. Ferreira, J.P.: Análise de estratégias de resposta a derramamento de óleo pesado no litoral do Espírito Santo utilizando modelagem computacional (in Portuguese). Msc. Dissertation, Programa de Pós-Graduação em Engenharia Ambiental, UFES (2006)
17. CEPEMAR: Centro de Pesquisas do Mar, Vitória, Brasil. EIA – Estudo de Impacto Ambiental para as Atividades de Produção e Escoamento de Petróleo do Campo de Jubarte – Fase 1 do Desenvolvimento da Produção. RT 017/04 (in Portuguese) (2004)
18. ASA: OILMAP – Oil Spill Model Assessment Package. Applied Science Associates (2004), www.appsci.com
19. Ji, S., Li, X., Du, R.: Tolerance Synthesis using Second-Order Fuzzy Comprehensive Evaluation and Genetic Algorithm. International Journal of Production Research 38, 3471–3483 (2000)

Sensor Fusion Map Building-Based on Fuzzy Logic Using Sonar and SIFT Measurements

Alfredo Chávez Plascencia and Jan Dimon Bendtsen

Abstract. This article presents a sensor data fusion method that can be used for map building. This takes into account the uncertainty inherent in sensor measurements. To this end, fuzzy logic operators are used to fuse the sensory information and to update the fuzzy logic maps. The sensory information is obtained from a sonar array and a stereo vision system. Features are extracted using the Scale Invariant Feature Transform (SIFT) algorithm. The approach is illustrated using actual measurements from a laboratory robot.

1 Introduction

In the field of autonomous mobile robots one of the main requirements is to have the capacity to operate independently in uncertain and unknown environments; fusion of sensory information and map building are some of the key capabilities that the mobile robot has to possess in order to achieve autonomy. Map building must be performed based on data from sensors; the data in turn must be interpreted and fused by means of sensor models. The result of the fusion of the sensor information is utilised to construct a map of the robot's environment. In this paper, a sensor data fusion application to map building is presented. The approach is exemplified by building a map for a laboratory robot by fusing range readings from a sonar array with landmarks extracted from stereo vision images using the SIFT algorithm.

Alfredo Chávez Plascencia
Department of Electronic Systems, Automation and Control, Fredrik bajers Vej 7C, 9229 Aalborg, Denmark
e-mail: acp@es.aau.dk

Jan Dimon Bendtsen
Department of Electronic Systems, Automation and Control, Fredrik bajers Vej 7C, 9229 Aalborg, Denmark
e-mail: dimon@es.aau.dk

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 13–22.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

Fuzzy set theory provides a natural framework in which uncertain information can be handled. Fuzzy sets are used to represent the uncertainty in sensor measurements. The resulting representation is similar to the occupancy grid, commonly obtained by stochastic theory [1, 2]. In the fuzzy approach, the environment is represented as a universal set $\mathscr{U}$, in which a real number is associated to each point quantifying the possibility that it belongs to an obstacle, [13]. Two fuzzy sets $\mathscr{O}$ and $\mathscr{E}$ that belong to $\mathscr{U}$ are defined to represent the evidence of a single cell $C_{i\,j}$ being occupied and empty respectively. These two sets are no longer complementary. In the map building process, the sensor data fusion approach and the map updating are carried out by means of fuzzy set operators. A thorough comparison between fuzzy vs. probabilistic is presented in [13]. The aim of this paper is to show the feasibility of *SIFT*-sonar map building based on fuzzy set operators.

2 Sensor Models

2.1 Sonar Model

A common sensor used to measure distance is the ultrasonic range finder, a.k.a. sonar. The sonar can measure the distance from the transducer to an object quite accurately. However, it cannot estimate at what angle within the sonar cone the pulse was reflected. Hence, there will be some uncertainty about the angle at which the obstacle was measured. A wide range of sonar models have been developed in the past years by various researchers, [1, 2, 3, 4]. For instance, [4] presents a modified version of the probabilistic Elfes-Moravec model, [1]. [14] combines the models presented in [1, 4]. This model in fact combines the quadratic and exponential distributions in the empty probability region of the sonar model. Due to its robustness this model is used in this paper.

Consider the representation of the sonar beam cone shown in figure 1, where the sonar beam is formulated as two functions. These functions measure the confidence and uncertainty of an empty and occupied region in the cone beam of the sonar respectively. They are defined based on the geometrical aspect and the spatial sensitivity of the sonar beam. Let Ψ denote the top angle of the cone in the horizontal plane and let ϕ denote the (unknown) angle from the centre line of the beam to the

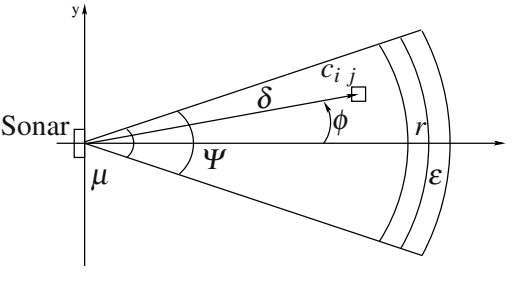


Fig. 1 Sonar Model

grid cell $C_{i\,j}$. Let r denote a sonar range measurement and ε the mean sonar deviation error. The value μ in the sonar model represents the minimal measurement and δ is the distance from the sonar to the cell.Then $f_s^e(\delta\ \phi\ r) = F_s(\delta\ r)A_n(\phi)$ represents the evidence of the cell $C_{i\,j}$ (translated from polar coordinates $(r\ \phi)$) being empty, and $f_s^o(\delta\ \phi\ r) = \mathscr{O}_s(\delta\ r)A_n(\phi)$ represents the evidence of the cell $C_{i\,j}$ being occupied. The factors F_s, $\mathscr{O}_s$ and $A_n(\phi)$ are given by the expressions stated in 1 as shown in [14].

$$
\begin{aligned}
F_s(\delta,r) &= \begin{cases} 1-\left(\frac{\delta}{r}\right)^2, & \text{if } \delta\in[0,\mu] \\ e^{\delta}, & \text{if } \delta\in[\mu,r-\varepsilon] \\ 0 & \text{otherwise} \end{cases} \\
\mathscr{O}_s(\delta,r) &= \begin{cases} \frac{1}{r}\left(1-\left(\frac{\delta-r}{\varepsilon}\right)^2\right), & \text{if } \delta\in[r-\varepsilon,r+\varepsilon] \\ 0 & \text{otherwise} \end{cases} \\
A_n(\phi) &= \begin{cases} 1-\left(\frac{2\phi}{\Psi}\right)^2, & \text{if } \phi\in\left[-\frac{\Psi}{2},\frac{\Psi}{2}\right] \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{1}
$$

2.2 Vision-SIFT-Descriptor Model

2.2.1 SIFT

The other sensor used for sensor fusion in this study is a stereo vision system. In particular, the Scale Invariant Feature Transform (SIFT) is a method for extracting distinctive invariant features from digital images [5]. The features are invariant to scaling and rotation. They also provide a robust matching across a substantial range of affine distortion, change in 3D view point, addition of noise and change in illumination. Furthermore, the features are distinctive, i.e. they can be matched with high probability to other features in a large database with many images. Once the descriptors are found in each image, i.e. left and right images, a matching algorithm is applied in both images. Figure 2 presents the matching feature descriptors which have been identified from a stereo pair of images.

2.3 SIFT-Descriptor Model

The triangulation algorithm outlined in [6] has been implemented in order to obtain the depth of the matching SIFT descriptors.

Due to the factors of quantification and calibration errors, a certain degree of uncertainty must be expected in the triangulation. Mathies and Shafer [11] shows how to model and calculate the triangulation error in stereo matched with $3D$ normal distributions. Geometrically these uncertainties translate into ellipsoidal regions.

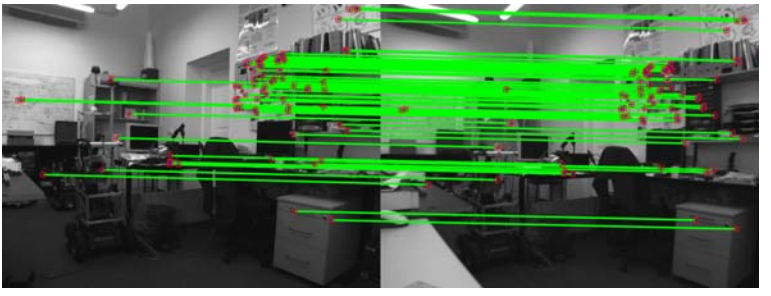


Fig. 2 Descriptor matches between left and right images

The stereo uncertainty error and the 3*D* Gaussian distribution can be depicted as in figures 3(a) and 3(b).

The empty regions from the left and right cameras as shown as shadow areas in figure 3(a) also need to be modelled. In [1], the empty area of the sonar model has a probabilistic representation. This approach has been taken into consideration and implemented with satisfactory results. Figure 4(a) shows a 3D model of the uncertainty triangulation together with the empty uncertainty region of the empty areas, which in fact is the 3D probability model of the SIFT-descriptor.

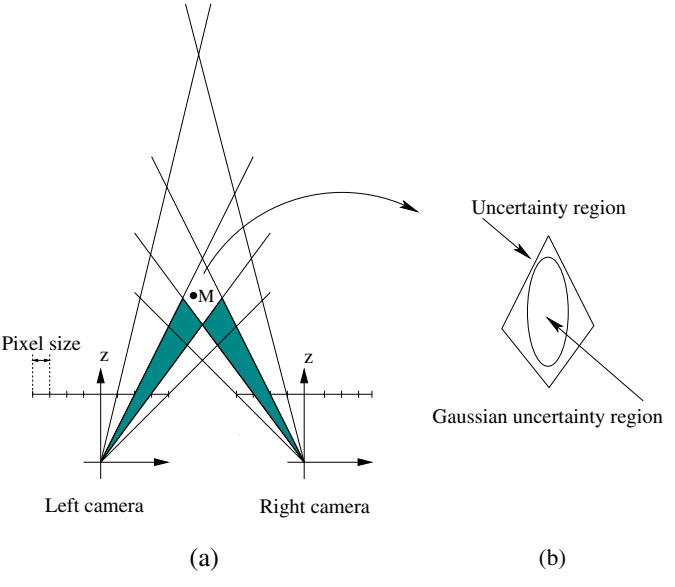
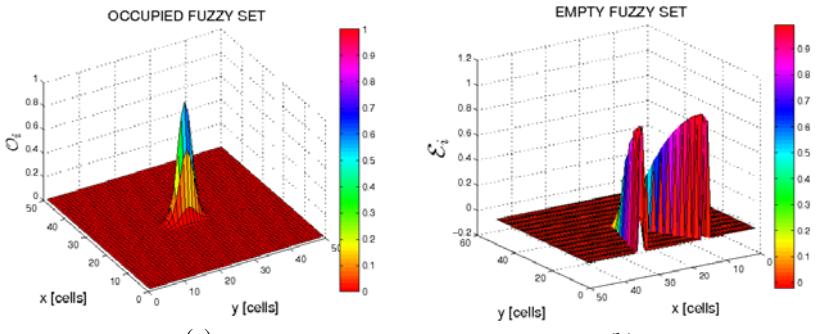


Fig. 3 ((a) Stereo Geometry showing triangulation uncertainty as a diamond around a point *M*. It also shows the empty region uncertainty from the pair of cameras to the uncertainty region of the point *M*. (b) 2D dimensional Gaussian distribution uncertainty region



(a) (b)

Fig. 4 (a) 3D representation of the occupied area by the *SIFT*-descriptor. (b) 3D representation of the empty area of the *SIFT*-descriptor

3 Sensor Fusion Based on Fuzzy Set Theory

3.1 Dombi Operator

Fuzzy logic offers a natural framework in which uncertain information can be handled [7]. The studies of fuzzy set theory can be traced back to the work done by Zadeh in the middle of $60's$ and early $70's$. Further information can be found in the references [8, 9].

A proper survey of fuzzy set operators for sensor data fusion can be found in [10]. The union Dombi fuzzy set operator, 2, was introduced by Dombi [12] and has been used in sensor data fusion. Due to its success and inspired by [7] the dombi operator is the choice of this paper.

$$u_\lambda(\mu_{\mathscr{A}}(x)\ \mu_{\mathscr{B}}(x)) = \frac{1}{1+\left(\frac{1}{\mu_{\mathscr{A}}(x)}-1\right)^{-\lambda}+\left(\frac{1}{\mu_{\mathscr{B}}(x)}-1\right)^{-\lambda}\Big)^{-\frac{1}{\lambda}}} \tag{2}$$

with $\lambda \in (0\ \infty)$.

The Dombi operator has the following property.

if $\lambda_1 \quad \lambda_1 \Longrightarrow u_{\lambda_1}(\mu_{\mathscr{A}}(x)\ \mu_{\mathscr{B}}(x)) \quad u_{\lambda_2}(\mu_{\mathscr{A}}(x)\ \mu_{\mathscr{B}}(x))$. This can be interpreted as λ is increased the Dombi operator produces larger union sets.

3.2 Fuzzy Maps from Sensor Measurements

Map building from sensor readings provides an example of a kind of uncertainty named lack of evidence, that does not indicate whether a given element is a member of a particular crisp set. Moreover, in the fuzzy context, the two sets $\mathscr{E}$ and $\mathscr{O}$ are no longer complementary: for a given point, partial membership to both $\mathscr{E}$ and $\mathscr{O}$ is possible, [13].

The fuzzy map building problem can be formulated as defining these two fuzzy sets $\mathscr{O}$ and $\mathscr{E}$ over a universal set $\mathscr{U}$. Such that, $\mathscr{O} \subset \mathscr{U} \subset \mathbb{R}^2$ and $\mathscr{E} \subset \mathscr{U} \subset \mathbb{R}^2$. The purpose of these two fuzzy sets is to allocate the evidence of a single cell $C_{i\,j}$ being occupied or empty by the degree of their respective membership functions $\mu_{\mathscr{O}}(x)$ and $\mu_{\mathscr{E}}(x)$ for each $x \in \mathscr{U}$. The degree of the membership functions stem from the interpreted sensor data by the sensor models. To this end, two local fuzzy sets $\mathscr{O}^k$ and $\mathscr{E}^k$ that contain local evidence are defined.

The sensor fusion and the map updating can be carried out by means of fuzzy set operators. The Dombi union operator is used to handle the sensor data fusion and the map updating. The local fuzzy maps $\mathscr{O}^k$ and $\mathscr{E}^k$ updates the global fuzzy maps $\mathscr{O}$ and $\mathscr{E}$ over the Dompi operator.

$$\begin{aligned} \mathscr{O} &= \mathscr{O} \cup \mathscr{O}^k \\ \mathscr{E} &= \mathscr{E} \cup \mathscr{E}^k \end{aligned} \tag{3}$$

A fuzzy map $\mathscr{M}$ that identifies unsafe cells by complementing a safe map $\mathscr{S}$ can be used for the robot during its motion.

$$\mathscr{M} = \bar{\mathscr{S}} \tag{4}$$

Where

$$\mathscr{S} = \mathscr{E}^2 \cap \bar{\mathscr{O}} \cap (\bar{\mathscr{E}} \cup \bar{\mathscr{O}}) \cap (\mathscr{E} \cup \mathscr{O}) \tag{5}$$

In the above formula, $\mathscr{S}$ is a conservative fuzzy map that identifies safe cells, and is obtained by subtracting from the very empty ones $\mathscr{E}^2$ the complement of the ambiguous, the indeterminate and the occupied sets. According to [13], by squaring the value of the membership function $\mathscr{E}$, the difference between low and high values is emphasised. The set of ambiguous cells is defined as $(\mathscr{A} = \mathscr{E} \cap \mathscr{O})$ and its complement is $(\bar{\mathscr{A}} = \bar{\mathscr{E}} \cup \bar{\mathscr{O}})$, whereas the set of indeterminate cells is defined as $(\mathscr{I} = \bar{\mathscr{E}} \cap \bar{\mathscr{O}})$ and its complement is $\bar{\mathscr{I}} = (\mathscr{E} \cup \mathscr{O})$.

The above fuzzy set computations are performed by the complement and the intersection operators 6.

$$\begin{aligned} c(\mu_{\mathscr{A}}(x)) &= 1 - \mu_{\mathscr{A}}(x) \\ i_1(\mu_{\mathscr{A}}(x)\ \mu_{\mathscr{B}}(x)) &= min(\mu_{\mathscr{A}}(x)\ \mu_{\mathscr{B}}(x)) \end{aligned} \tag{6}$$

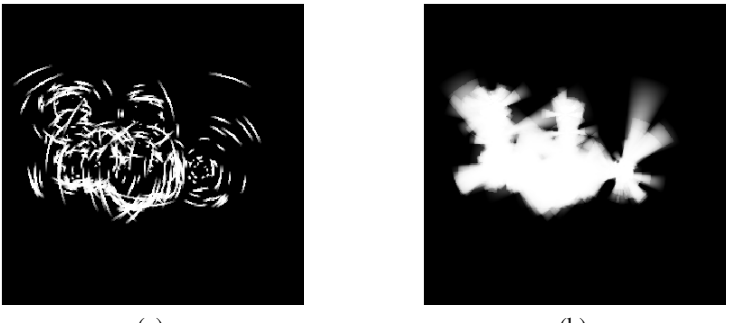
4 Map Building Experimental Results

A Pioneer3AT from ActiveMedia serves as an experimental testbed. It provides data by a ring of 16 ultrasonic sensors, stereo vision system and a laser rangefinder. The laser rangefinder was used for the purpose of evaluating the incoming data

from the sonar and the stereo pair of cameras respectively. The experiment was carried out in a typical office/laboratory environment. In this environment, the robot travels a random trajectory and collects data. The mobile robot makes a total of $n = 30$ measurements along this trajectory. In each measurement the robot scans the environment of the laboratory and gathers the data by means of the mentioned sensors. The laser gets 361 readings in the interval $[0^o\ 180^o]$. The sonar ring scan the environment in the interval $[0^o\ 360^o]$. The vision system receives the features from the overlapping field of view of the two cameras. The ring of sonars is placed around the robot, the vision and laser systems are aligned vertically over the sonar ring and they are placed in the front of the robot.

Figure 5(a) and 5(b) show the global $\mathcal{O}$ fuzzy set maps based on sonar readings. The ambiguous sonar fuzzy set map is presented in figure 6(c). Finally, the fuzzy sonar map $\mathcal{M}$ is depicted in figure 6(d).

Sensor fusion map building experiments based on fuzzy logic set operators using stereo vision system are shown in figure 7. More precisely, figure 7(a) and 7(b) depicts the global $\mathcal{O}$ fuzzy set maps. Figure 8(c) presents the ambiguous vision fuzzy set map. Finally, the fuzzy vision map $\mathcal{M}$ is depicted in figure 8(d).



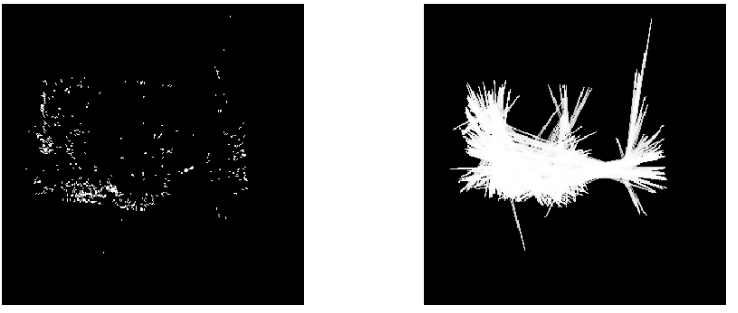
(a) (b)

Fig. 5 (a) Sonar occupied fuzzy set map (b) Sonar empty fuzzy set map



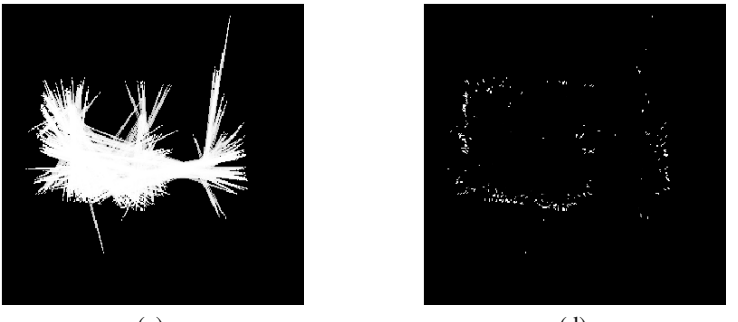
(c) (d)

Fig. 6 (c) Ambiguous sonar fuzzy set map (d) Fuzzy sonar map



(a) (b)

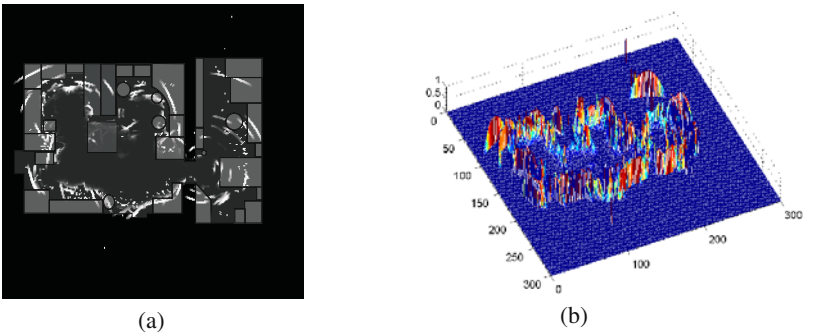
Fig. 7 (a) Vision occupied fuzzy set map (b) Vision empty fuzzy set map



(c) (d)

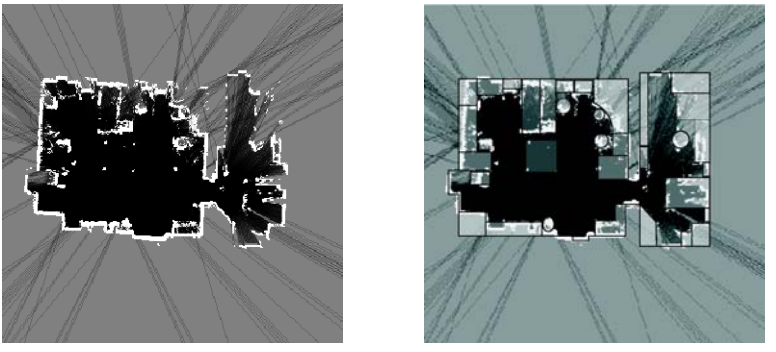
Fig. 8 (c) Ambiguous vision fuzzy set map (d) Fuzzy vision map

The intersection of the two global fuzzy set maps, $\mathcal{O}$ and $\mathcal{E}$, can be seen in figure 9(a) and 9(b). Note the satisfactory accordance of the map with the shape of the laboratory/office. The empty area is defined quite well even though the amount of



(a) (b)

Fig. 9 Intersection of the two global fuzzy set maps (a) 2*D* representation (b) 3*D* representation



(a) (b)

Fig. 10 (a) Represents the map of the laboratory/office based laser readings. (b) The map of the laboratory/office embedded into the laser map

measurements were not abundant enough to create a dense map. The former can better be seen by taking a look at the figure 9(b).

Figure 10(a) shows the grid created only from laser rangefinder data. This picture demonstrates the room shape. The layout of the laboratory/office is embedded into the laser map as seen in figure 10(b). It can be seen that the laser map is quite accurate when comparing it with the layout of the laboratory/office, and for this reason it is taken reference map.

5 Conclusion

Experimental results have shown the feasibility of the use of fuzzy set operators in map building based on interpreted sensor data fusion readings. They have also illustrated satisfactory performance in the applicability of fuzzy logic in the integration of the *SIFT* algorithm in the area of sensor fusion in mobile robots.

References

1. Moravec, H., Elfes, A.E.: High Resolution Maps from Wide Angle Sonar. In: Proceedings of the 1985 IEEE International Conference on Robotics and Automation, pp. 116–121 (1985)
2. Elfes, A.: Using Occupancy Grids for Mobile Robot Perception and Navigation. IEEE Journal of Robotics and Automation 22, 45–57 (1989)
3. Konolige, K.: Improved Occupancy Grids for Map Building. Autonomous Robots 4, 351–367 (1997)
4. Štěpán, P., Přeučil, L., Král, L.: Statistical Approach to Integration Interpretation of Robot Sensor Data. In: Proceedings of the IEEE International Workshop on Expert Systems Applications, pp. 742–747 (1997)

5. Lowe, G.: Distinctive Image Features from Scale-Invariant Key points. International Journal of Computer Vision 60, 91–110 (2004)
6. Trucco, E., Verri, A.: Introductory techniques for 3-D Computer Vision. Prentice Hall, Englewood Cliffs (1998)
7. Oriolo, G., Ulivi, G., Vendittelli, M.: Real-time map building and navigation for autonomous robots in unknown environments. IEEE Transactions on Systems, Man, and Cybernetics, Part B 28, 316–333 (1998)
8. Timothy, J.: Fuzzy Logic with Engineering Applications. McGraw-Hill, University of New Mexico (1995)
9. Klir, G.J., Folger, T.A.: The Grid: Fuzzy Sets, Uncertainty and Information. Prentice Hall, Englewood Cliffs (1998)
10. Bloch, I.: Information Combination Operators for Data Fusion: A Comparative review with Classification. IEEE Transactions of Systems, Man and Cybernetics 26(1), 52–67 (1996)
11. Mathies, L.: Error Modeling in Stereo Navigation. IEEE Journal of Robotics and Automation 3(3), 239–248 (1987)
12. Dombi, J.: A general class of fuzzy operators, the De Morgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. Fuzzy Sets and Systems 8, 149–163 (1982)
13. Oriolo, G., Ulivi, G., Vendittelli, M.: Fuzzy Maps: A New Tool for Mobile Robot Perception and Planning. Journal of Robotic Systems 14, 179–197 (1997)
14. Chávez, A., Stepan, P.: Sensor Data Fusion. In: Proceedings of the IEEE Conference on Advances in Cybernetics Systems, pp. 20–25. Sheffiel University, United Kingdom (2006)

Rough Sets in Medical Informatics Applications

Aboul Ella Hassanien, Ajith Abraham, James F. Peters, and Gerald Schaefer

Abstract. Rough sets offer an effective approach of managing uncertainties and can be employed for tasks such as data dependency analysis, feature identification, dimensionality reduction, and pattern classification. As these tasks are common in many medical applications it is only natural that rough sets, despite their relative 'youth' compared to other techniques, provide a suitable method in such applications. In this paper, we provide a short summary on the use of rough sets in the medical informatics domain, focussing on applications of medical image segmentation, pattern classification and computer assisted medical decision making.

1 Introduction

Rough set theory provides an approach to approximation of sets that leads to useful forms of granular computing. The underlying concept is to extract to what extent a

Aboul Ella Hassanien
Information Technology Department, FCI, Cairo University, and the System Department, CBA, Kuwait University, Kuwait
e-mail: abo@cba.edu.kw

Ajith Abraham
Center for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, Trondheim, Norway
e-mail: ajith.abraham@ieee.org

James F. Peters
Computational Intelligence Laboratory, Department of Electrical & Computer Engineering, University of Manitoba, Winnipeg, Canada
e-mail: jfpeters@ee.umanitoba.ca

Gerald Schaefer
School of Engineering and Applied Science, Aston University Birmingham, U.K.
e-mail: g.schaefer@aston.ac.uk

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 23–30.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

given set of objects (e.g. extracted feature samples) approximate another set of objects of interest. Rough sets offer an effective approach of managing uncertainties and can be employed for tasks such as data dependency analysis, feature identification, dimensionality reduction, and pattern classification.

Based on rough set theory it is possible to construct a set of simple if-then rules from information tables. Often, these rules can reveal previously undiscovered patterns in sample data. Rough set methods can also be used to classify unknown data based on already gained knowledge. Unlike many other techniques, rough set analysis requires no external parameters and uses only the information present in the input data. Rough set theory can be utilised to determine whether sufficient data for a task is available respectively to extract a minimal sufficient set of features for classification which in turn effectively performs feature space dimensionality reduction.

Although, compared to other methods, a relatively recent technique, these characteristics have prompted various rough set approaches in the general domain of medical informatics. In the following we will therefore, after giving a brief introduction to basic rough set concepts, provide an overview of the use of rough sets in this area. In particular, we will show how rough sets have been used for medical image segmentation, classification, for mining medical data, and in medical decision support systems.

2 Rough Set Theory

Rough set theory [11, 14] is a fairly recent intelligent technique for managing uncertainty that is used for the discovery of data dependencies, to evaluate the importance of attributes, to discover patterns in data, to reduce redundancies, and to recognise and classify objects. Moreover, it is being used for the extraction of rules from databases where one advantage is the creation of readable if-then rules. Such rules have the potential to reveal previously undiscovered patterns in the data; furthermore, it also collectively functions as a classifier for unseen samples. Unlike other computational intelligence techniques, rough set analysis requires no external parameters and uses only the information presented in the given data. One of the useful features of rough set theory is that it can tell whether the data is complete or not based on the data itself. If the data is incomplete, it will suggest that more information about the objects is required. On the other hand, if the data is complete, rough sets are able to determine whether there are any redundancies and find the minimum data needed for classification. This property of rough sets is very important for applications where domain knowledge is very limited or data collection is expensive because it makes sure the data collected is just sufficient to build a good classification model without sacrificing accuracy [11, 14].

In rough set theory, sample objects of interest are usually represented by a table called an information table. Rows of an information table correspond to objects and columns correspond to object features. For a given set B of functions representing object features and a set of sample objects X, an indiscernibility relation $\sim_B$ is a set of pairs $(x\ x') \in X \times X$ such that $f(x) = f(x')$ for all $f \in B$. The relation $\sim_B$ defines a

quotient set $X \sim_B$, i.e., a set of all classes in the partition of X defined by $\sim_B$. Rough set theory identifies three approximation regions defined relative to $X \sim_B$, namely, lower approximation, upper approximation and boundary. The lower approximation of a set X contains all classes that are subsets of X, the upper approximation contains all classes with non-empty intersections with X, and the boundary is the set difference between the upper and lower approximations.

Rough image processing can be defined as the collection of approaches and techniques that understand, represent and process images, their segments and features as rough sets [21]. In images boundaries between object regions are often ill-defined [10]. This uncertainty can be handled by describing the different objects as rough sets with upper (or outer) and lower (or inner) approximations.

3 Rough Sets in Medical Image Segmentation

One of the most important tasks in medical imaging is segmentation as it is often a pre-cursor to subsequent analysis, whether manual or automated. The basic idea behind segmentation-based rough sets is that while some cases may be clearly labelled as being in a set X (called positive region in rough sets theory), and some cases may be clearly labelled as not being in X (called negative region), limited information prevents us from labelling all possible cases clearly. The remaining cases cannot be distinguished and lie in what is known as the boundary region. Kobashi *et al.* [7] introduced rough sets to treat nominal data based on concepts of categorisation and approximation for medical image segmentation. The proposed clustering method extracts features of each pixel by using thresholding and labelling algorithms. Thus, the features are given by nominal data. The ability of the proposed method was evaluated by applying it to human brain MRI images. Peters *et al.* [12] presented a new form of indiscernibility relation based on k-means clustering of pixel values. The end result is a partitioning of a set of pixel values into bins that represent equivalence classes. The proposed approach allows to introduce a form of upper and lower approximation specialised relative to sets of pixel values.

An improved clustering algorithm based on rough sets and entropy theory was presented by Chena and Wang [1]. The method avoids the need to pre-specify the number of clusters which is a common problem in clustering based segmentation approaches. Clustering can be performed in both numerical and nominal feature spaces with a similarity introduced to replace the distance index. At the same time, rough sets are used to enhance the algorithm with the capability to deal with vagueness and uncertainty in data analysis. Shannon's entropy was used to refine the clustering results by assigning relative weights to the set of features according to the mutual entropy values. A novel measure of clustering quality was also presented to evaluate the clusters. The experimental results confirm that both efficiency and clustering quality of this algorithm are improved.

An interesting strategy for colour image segmentation using rough sets has been presented by Mohabey *et al.* [9]. They introduced a concept of encrustation of the histogram, called histon, for the visualisation of multi-dimensional colour information

in an integrated fashion and study its applicability in boundary region analysis. The histon correlates with the upper approximation of a set such that all elements belonging to this set are classified as possibly belonging to the same segment or segments showing similar colour value. The proposed encrustation provides a direct means of separating a pool of inhomogeneous regions into its components. This approach can then be extended to build a hybrid rough set theoretic approximations with fuzzy c-means based colour image segmentation. The technique extracts colour information regarding the number of segments and the segment centers of the image through rough set theoretic approximations which then serve as the input to a fuzzy c-means algorithm.

Widz *et al.* [20] introduced an automated multi-spectral MRI segmentation technique based on approximate reducts derived from the theory of rough sets. They utilised T1, T2 and PD MRI images from a simulated brain database as a gold standard to train and test their segmentation algorithm. The results suggest that approximate reducts, used alone or in combination with other classification methods, may provide a novel and efficient approach to the segmentation of volumetric MRI data sets. Segmentation accuracy reaches 96% for the highest resolution images and 89% for the noisiest image volume. They tested the resultant classifier on real clinical data, which yielded an accuracy of approximately 84%.

4 Rough Sets in Medical Classification

The computation of the core and reducts from a rough set decision table is a way of selecting relevant features [15]. It is a global method in the sense that the resultant reducts represent the minimal sets of features which are necessary to maintain the same classification power given by the original and complete set of features. A more direct manner for selecting relevant features is to assign a measure of relevance to each feature and choose the features with higher values. Based on the reduct system, we can generate the list of rules that will be used for building the classifier model for the new objects. Reduct is an important concept in rough set theory and data reduction is a main application of rough set theory in pattern recognition and data mining.

Wojcik [21] approached the nature of a feature recognition process through the description of image features in terms of rough sets. Since the basic condition for representing images must be satisfied by any recognition result, elementary features are defined as equivalence classes of possible occurrences of specific fragments existing in images. The names of the equivalence classes (defined through specific numbers of objects and numbers of background parts covered by a window) constitute the best lower approximation of the window contents (i.e., names of recognised features). The best upper approximation is formed by the best lower approximation, its features, and parameters, all referenced to the object fragments located within the window. The rough approximation of shapes is robust with respect to accidental changes in the width of contours and lines and to small discontinuities and, in

general, to possible positions or changes in shape of the same feature. Rough sets are also used for noiseless image quantisation.

Swiniarski and Skowron [16] presented applications of rough set methods for feature selection in pattern recognition. They emphasise the role of basic constructs of rough set approaches in feature selection, namely reducts and their approximations, including dynamic reducts. Their algorithm for feature selection is based on the application of a rough set method to the result of principal component analysis (PCA) used for feature projection and reduction. In their study, mammogram images were evaluated for recognition experiments. The database contains three types of images: normal, benign, and malignant. For each abnormal image the co-ordinates of centre of abnormality and proximate radius (in pixels) of a circle enclosing the abnormality, have been given. For classification the centre locations and radii apply to clusters rather than to the individual classifications. From the original mammograms, 64 x 64 pixel sub-images were extracted around the center of abnormality (or at the average co-ordinate for normal cases). They concluded that the rough set methods have shown ability to significantly reduce the pattern dimensionality and have proven to be viable image mining techniques as a front end of neural network classifiers.

Cyran and Mrzek [2] showed how rough sets can be applied to improve the classification ability of a hybrid pattern recognition system. Their system consists of a feature extractor based on a computer-generated hologram (CGH) where the extracted features are shift, rotation, and scale invariant. An original method of optimising the feature extraction abilities of a CGH was introduced which uses rough set concepts to measure the amount of essential information contained in the feature vector. This measure is used to define an objective function in the optimisation process. Since rough set based factors are not differentiable, they use a nongradient approach for a search in the space of possible solutions. Finally, rough sets are used to determine decision rules for the classification of feature vectors.

5 Rough Sets in Medical Data Mining

With increasing sizes of the amount of data stored in medical databases, efficient and effective techniques for medical data mining are highly sought after. Applications of rough sets in this domain include inducing propositional rules from databases using rough sets prior to using these rules in an expert system. Tsumoto [18] presented a knowledge discovery system based on rough sets and feature-oriented generalisation and its application to medicine. Diagnostic rules and information on features are extracted from clinical databases on diseases of congenital anomaly. Experimental results showed that the proposed method extracts expert knowledge correctly and also discovers that symptoms observed in six positions (eyes, noses, ears, lips, fingers, and feet) play important roles in differential diagnosis.

Hassanien *el al.* [4] presented a rough set approach to feature reduction and generation of classification rules from a set of medical datasets. They introduced a rough set reduction technique to find all reducts of the data that contain the minimal subset of features associated with a class label for classification. To evaluate the validity of

the rules based on the approximation quality of the features, a statistical test to evaluate the significance of the rules was introduced. A set of data samples of patients with suspected breast cancer were used and evaluated. The rough set classification accuracy was shown to compare favourably with the well-known ID3 classifier algorithm.

Huang and Zhang [5] presented a new application of rough sets to ECG recognition. First, the recognition rules for characteristic points in ECG are reduced using rough set theory. Then the reduced rules are used as restriction conditions of an eigenvalue determination arithmetic to recognise characteristic points in ECG. Several aspects of correlative arithmetic such as sizer method, difference method and how to choose difference parameters are discussed. They also adopted MIT-BIH data to verify R wave recognition and it is shown that the resulting detection rate is higher than those of conventional recognition methods.

Recently, Independent Component Analysis (ICA) [6] has gained popularity as an effective method for discovering statistically independent variables (sources) for blind source separation, as well as for feature extraction. Swiniarski *et al.* [17] studied several hybrid methods for feature extraction/reduction, feature selection, and classifier design for breast cancer recognition in mammograms. The methods included independent component analysis, principal component analysis (PCA) and rough set theory. Three classifiers were designed and tested: a rough sets rule-based classifier, an error back propagation neural network, and a Learning Vector Quantization neural network. Based on a comparative study on two different data sets of mammograms, rough sets rule-based classifier performed with a significantly better level of accuracy than the other classifiers. Therefore, the use of ICA or PCA as a feature extraction technique in combination with rough sets for feature selection and rule-based classification offers an improved solution for mammogram recognition in the detection of breast cancer.

6 Rough Sets in Medical Decision Support Systems

The medical diagnosis process can be interpreted as a decision-making process, during which the physician induces the diagnosis of a new and unknown case from an available set of clinical data and from clinical experience. This process can be computerised in order to present medical diagnostic procedures in a rational, objective, accurate and fast way. In fact, during the last two or three decades, diagnostic decision support systems have become a well-established component of medical technology.

Podraza *et. al* [13] presented an idea of complex data analysis and decision support system for medical staff based on rough set theory. The main aim of their system is to provide an easy to use, commonly available tool for efficiently diagnosing diseases, suggesting possible further treatment and deriving unknown dependencies between different data coming from various patient's examinations. A blueprint of a possible architecture of such a system is presented including some example algorithms and suggested solutions, which may be applied during implementation.

The unique feature of the system relies on removing some data through rough set decisions to enhance the quality of the generated rules. Usually such data is discarded, because it does not contribute to the knowledge acquisition task or even hinder it. In their approach, improper data (excluded from the data used for drawing conclusions) is carefully taken into considerations. This methodology can be very important in medical applications as a case not fitting to the general classification cannot be neglected, but should be examined with special care.

Mitra *et al.* [8] implemented a rule-based rough-set decision system for the development of a disease inference engine for ECG classification. ECG signals may be corrupted by various types of noise. Therefore, at first, the extracted signals are undergoing a noise removal stage. A QRS detector is also developed for the detection of R-R interval of ECG waves. After the detection of this R-R interval, the P and T waves are detected based on a syntactic approach. Isoelectric-level detection and base-line correction are also implemented for accurate computation of different features of P, QRS, and T waves. A knowledge base is developed from medical literature and feedback of reputed cardiologists regarding ECG interpretation and essential time-domain features of the ECG signal. Finally, a rule-based rough-set decision system is generated for the development of an inference engine for disease identification from these time-domain features.

Wakulicz-Deja and Paszek [19] implemented an application of rough set theory to decision making for diagnosing mitochondrial encephalomyopathies in children. The resulting decision support system maximally limits the indications for invasive diagnostic methods (puncture, muscle and/or nerve specimens). Moreover, it shortens the time necessary for making diagnosis. The system has been developed on the basis of data obtained from the Clinic Department of Pediatrics of the Silesian Academy of Medicine.

7 Conclusions

In this paper, we have provided a brief overview of rough sets and their use in various medical tasks. Although rough sets represent a relatively recent approach, a number of effective applications have demonstrated their potential and it is only to be expected that research will continue to improve upon and extend these techniques. Due to space constraints we were only able to highlight some of the work on medical imaging and medical decision making. A more comprehensive review of the literature on rough sets in medical imaging can be found in [3].

References

1. Chena, C.-B., Wang, L.-Y.: Rough set-based clustering with refinement using shannon's entropy theory. Computers and Mathematics with Applications 52(10-11), 1563–1576 (2006)
2. Cyran, K.A., Mrzek, A.: Rough sets in hybrid methods for pattern recognition. International Journal of Intelligent Systems 16(2), 149–168 (2001)

3. Hassanien, A.E., Abraham, A., Peters, J.F., Schaefer, G.: Overview of rough-hybrid approaches in image processing. In: IEEE Conference on Fuzzy Systems, pp. 2135–2142 (2008)
4. Hassanien, A.E., Ali, J.M., Hajime, N.: Detection of spiculated masses in mammograms based on fuzzy image processing. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 1002–1007. Springer, Heidelberg (2004)
5. Huang, X.-M., Zhang, Y.-H.: A new application of rough set to ECG recognition. In: Int. Conference on Machine Learning and Cybernetics, vol. 3, pp. 1729–1734 (2003)
6. Hyvärinen, A., Oja, E.: Independent component analysis: A tutorial. Technical report, Laboratory of Computer and Information Science, Helsinki University of Technology (1999)
7. Kobashi, S., Kondo, K., Hata, Y.: Rough sets based medical image segmentation with connectedness. In: 5th Int. Forum on Multimedia and Image Processing, pp. 197–202 (2004)
8. Mitra, S., Mitra, M., Chaudhuri, B.B.: A rough-set-based inference engine for ECG classification. IEEE Trans. on Instrumentation and Measurement 55(6), 2198–2206 (2006)
9. Mohabey, A., Ray, A.K.: Fusion of rough set theoretic approximations and FCM for color image segmentation. In: IEEE Int. Conference on Systems, Man, and Cybernetics, vol. 2, pp. 1529–1534 (2000)
10. Pal, S.K., Pal, B.U., Mitra, P.: Granular computing, rough entropy and object extraction. Pattern Recognition Letters 26(16), 2509–2517 (2005)
11. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning About Data. Kluwer, The Netherlands (1991)
12. Peters, J.F., Borkowski, M.: K-means indiscernibility relation over pixels. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS, vol. 3066, pp. 580–585. Springer, Heidelberg (2004)
13. Podraza, R., Dominik, A., Walkiewicz, M.: Decision support system for medical applications. In: Applied Simulation and Modelling (2003)
14. Polkowski, L.: Rough Sets. Mathematical Foundations. Physica-Verlag, Heidelberg (2003)
15. Ślęzak, D.: Various approaches to reasoning with frequency-based decision reducts: a survey. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) Rough Sets in Soft Computing and Knowledge Discovery: New Developments. Physica Verlag, Heidelberg (2000)
16. Swiniarski, R., Skowron, A.: Rough set methods in feature selection and recognition. Pattern Recognition Letters 24, 833–849 (2003)
17. Swiniarski, R.W., Lim, H.J., Shin, Y.H., Skowron, A.: Independent component analysis, princpal component analysis and rough sets in hybrid mammogram classification. In: Int. Conference on Image Processing, Computer Vision, and Pattern Recognition, p. 640 (2006)
18. Tsumoto, S.: Mining diagnostic rules from clinical databases using rough sets and medical diagnostic model. Information Sciences: an International Journal 162(2), 65–80 (2004)
19. Wakulicz-Deja, A., Paszek, P.: Applying rough set theory to multi stage medical diagnosing. Fundamenta Informaticae 54(4), 387–408 (2003)
20. Widz, S., Revett, K., Ślęzak, D.: Application of rough set based dynamic parameter optimization to mri segmentation. In: 23rd Int. Conference of the North American Fuzzy Information Processing Society, pp. 440–445 (2004)
21. Wojcik, Z.: Rough approximation of shapes in pattern recognition. Computer Vision, Graphics, and Image Processing 40, 228–249 (1987)

A Real Estate Management System Based on Soft Computing

Carlos D. Barranco, Jesús R. Campaña, and Juan M. Medina

Abstract. The paper describes a web based system which applies Soft Computing techniques to the area of real estate management. The application is built on a Fuzzy Object Relational Database Management System called Soft Data Server, which provides application capabilities for fuzzy data handling. A brief overview of fuzzy types and operations available in Soft Data Server is also depicted. The paper shows the way real estate attributes can be expressed using fuzzy data, and how fuzzy queries can be used to express typical real estate customer requirements on a fuzzy real estate database. Finally, a brief overview of the layered architecture of the application is presented.

1 Introduction

ImmoSoftDataServerWeb (ISDSW) is a web based application which takes advantage of fuzzy set theory applying it to the area of real estate management. The application is built on a Fuzzy Object Relational Database Management System (FORDBMS) which provides capabilities for fuzzy data handling. Real estate attributes are expressed using fuzzy data, and fuzzy queries are defined to obtain appropriate results.

In real estate brokerage, the search process is affected by vagueness due to flexibility in the search conditions. For instance, let us consider an usual case of a customer looking for an apartment within a price range. If the real estate sales agent

Carlos D. Barranco
Division of Computer Science, School of Engineering, Pablo de Olavide University, Utrera Rd. Km. 1, 41013 Sevilla, Spain
e-mail: cbarranco@upo.es

Jesús R. Campaña · Juan M. Medina
Dept. of Computer Science and Artificial Intelligence, University of Granada, Daniel Saucedo Aranda s/n, 18071 Granada, Spain
e-mail: {jesuscg,medina}@decsai.ugr.es

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 31–40.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

does not have an offer of an *apartment* but has an offer of a *flat* in this price range, the agent may consider informing the buyer about this offer, as it could be interesting for him. In this example, the condition $kind = apartment$ is made flexible in order to match buyer requirements to existing offers. The use of flexible conditions increases business opportunities by offering the customer a wider range of possible interesting real estates.

ISDSW provides flexible search capabilities, understanding flexible search as the definition of search conditions using a language closer to the user. This approach allows the obtained results to fit better the specified conditions, obtaining a more intelligent response to search requests. It also supports the visual definition of geographic conditions, which allow users to search for real estates located in a particular geographic zone. The interface for geographic conditions definition is very simple and intuitive for the user.

ISDSW aims to substitute real estate sales agents by an automated process integrated in a web application. To emulate the real estate agent flexibility, and to increase the buyer expression capabilities, ISDSW offer a wide variety of flexible conditions based on fuzzy set theory.

ISDSW is based on ImmoSoftWeb (ISW) [4], a web based real estate management application that provides flexible real estate search features to users. While ISW relies in FSQL Server [6], a fuzzy Relational DBMS server, ISDSW uses Soft Data Sever, a server providing fuzzy data management functionalities using Object Relational DBMS technologies.

The paper is organized as follows. Section 2 presents Soft Data Server and its features. Section 3 depicts ImmoSoftDataServerWeb, where application specifics and architecture are presented. Finally, Section 4 presents concluding remarks and future work.

2 Soft Data Server (SDS)

Soft Data Server (SDS) [2, 3, 5] is an extension of *Oracle*®, a well-known and widespread commercial ORDBMS.

SDS extends the host ORDBMS by taking advantage of the extension mechanisms included in the latest SQL standards, SQL:1999 [8] and SQL:2003 [7], as the underlying ORDBMS is compliant with useful parts of these standards. This extension allows to create a FORDBMS on top of the underlying ORDBMS.

SDS mainly defines a group of User Defined Types (UDTs) which hold the representation and manipulation details of fuzzy data in the database, and help the user to create his own fuzzy types.

These UDTs and their supertype/subtype relation are depicted in Figure 1. The figure includes a pair of abstract data types that do not correspond to real database types, but help to clarify the SDS type structure. The mentioned abstract data types are *Database Data Types*, which model a root type for every database data type, and *Built-In Types*, which is the common ancestor for the built-in data types of the

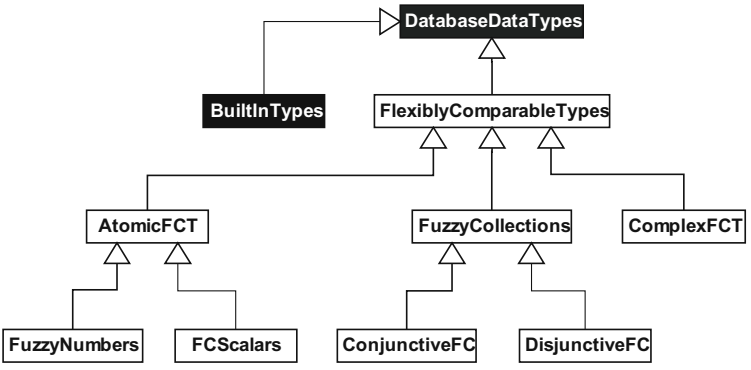


Fig. 1 SDS Datatype Hierarchy

ORDBMS. In the figure these abstract data types appear in a dark background in order to differentiate them from non-abstract UDTs.

The UDTs for fuzzy data representation and manipulation included in SDS are the following:

- *Flexibly Comparable Types (FCTs)*: This UDT is the common ancestor for all the UDTs included in SDS. Its main purpose is to encapsulate all the common and compulsory behavior for FCTs. One of these compulsory methods is an abstract method for flexible comparison (*feq*), which is redefined in each subtype in order to implement its corresponding flexible equivalence relation. This redefinition is particularly important for user defined FCTs, as the user attaches to the data type an especially designed flexible equivalence relation. Another common behavior encapsulated in this UDT is the set of methods to allow the definition, update and deletion of linguistic labels for the type.

  The *feq* method (Fuzzy Equal) is defined as in (1), where D is the domain defined as a FCT, and μ_{ER_D} is the membership function of the flexible equivalence relation defined for the FCT.

$$feq(a\ b) = \mu_{ER_D}(a\ b); a\ b \in D \tag{1}$$

  The domains (UDTs and built-in types) of SDS are divided into two separate groups, those implementing the *feq* method, and those not implementing it. The domains of the former type are named FCT types, and the domains of the latter group are named Non FTC (NFCT) types.
- *Atomic FCT*: This UDT acts as the common ancestor for those SDS UDTs designed to represent non complex or set oriented data, namely atomic fuzzy data, as fuzzy numbers and scalars.
- *Fuzzy Numbers*: This UDT is designed to represent and manage fuzzy numbers in SDS. This FCT models a domain whose elements are fuzzy numbers defined as trapezoidal possibility distributions. These distributions are modelled as four numerical built-in types $\mu_{[\alpha\ \beta\ \gamma\ \delta]}$. Using this representation it is possible

describe crisp values, approximate values, intervals and trapezoidal distributions. Trapezoidal possibility distributions are defined as shown in (2).

$$\mu_{[\alpha\ \beta\ \gamma\ \delta]}(x) = \begin{array}{ll} 0 & x \leq \alpha \text{ or } x \geq \delta \\ \frac{x-\alpha}{\beta-\alpha} & \alpha \quad x \quad \beta \\ 1 & \beta \leq x \leq \gamma \\ \frac{\delta-x}{\delta-\gamma} & \gamma \quad x \quad \delta \end{array} \qquad \alpha \leq \beta \leq \gamma \leq \delta \tag{2}$$

Additionally, this UDT includes methods implementing fuzzy relational comparators for fuzzy numbers. These fuzzy comparators are similar to those included in FSQL Server. The resemblance of two fuzzy numbers could be calculated by means of their possibility measure. Therefore, the *feq* method of this FCT can be implemented as shown in (3), where a and b are two fuzzy numbers, and $\otimes$ a t-norm.

$$feq(a\ b) = \sup_x(\mu_a(x) \otimes \mu_b(x)) \tag{3}$$

- *FC Scalars*: This UDT is designed as a common ancestor for the FCT representing discrete scalar domains where a flexible equivalence relation is defined. The type encloses helper methods for the definition and removal of FCTs representing flexibly comparable scalar domains along with their associated flexible equivalence relations. Each object of this FCT is able to represent one scalar, or less formally a label, of the domain.
- *Fuzzy Collections*: This UDT is the ancestor of every type included in SDS which represents an extension of the collection types of the ORDBMSs.
- *Conjunctive FC*: This UDT gathers all the necessary functionality related to fuzzy collections with conjunctive semantics. This functionality includes helper methods for the definition and deletion of fuzzy collections which base type is determined by the user.
- *Disjunctive FC*: This UDT is analogous to the previously described UDT, but with disjunctive semantics. As the previous UDT, this data type includes helper methods for the management of fuzzy collections.
- *Complex FCT*: This UDT is a common ancestor for every user defined FCT designed to represent and manage complex data organized as a structure of fields. It includes a set of methods that encapsulate the user defined behavior for the data type. This supertype includes a specific implementation of the *feq* method that is based on a generic flexible equivalence relation for flexibly comparable complex data types.
- *Special Linguistic Labels*: Every FCT has a predefined set of special linguistic labels. These linguistic labels represent special values which are used to model the ignorance of a field value, the UNKNOWN label, the inapplicability of a field, the UNDEFINED label, and the ignorance about the applicability of the field and its value if the field were applicable, the NULL label. These special linguistic labels were defined in the GEFRED model [10]. The definition of these labels is slightly different in SDS due to the use of complex constructs like collections.

Special linguistic labels in SDS are defined in terms of their resemblance values when they are compared to other domain values as is shown in (4).

$$\begin{aligned} &feq(d \text{ UNKNOWN}) = 1 \ \forall d \in D \\ &feq(d \text{ UNDEFINED}) = 0 \ \forall d \in D \cup \{\text{UNKNOWN}\} \\ &feq(d \text{ NULL}) = null \ \forall d \in D \cup \{\text{UNKNOWN UNDEFINED}\} \end{aligned} \tag{4}$$

2.1 Fuzzy Data Comparison

Even though the flexible equivalence relation of a user defined FCT can be any of the designed by the user to meet particular requirements, this section introduces a generic purpose flexible equivalence relation for user defined complex FCT.

The proposed flexible equivalence relation is a fuzzy resemblance measure for complex objects, whose attributes either are of FCT or NFCT types. For the sake of simplicity, the proposed flexible equivalence relation is named Complex Object Resemblance Measure (CORM). The original idea of CORM was first proposed in [9] for OODBMS context, and later adapted to the object-relational paradigm in [5].

The way CORM determines the resemblance between two objects of the same type is sketched in Fig. 2, where o_1 and o_2 are objects of the same FCT, a_1 a_2 a_n are attributes of these objects, and v_{ij} is the value of the attribute a_j for the object o_i. This procedure is divided in two steps. First, a resemblance degree between the pair of values of each attribute of the compared object is calculated. For the i-th attribute, the function $S_{ai}(o_1\ o_2)$ is used. Then, the resemblance degrees of each pair of attribute values are aggregated. For this purpose, the V_Q aggregator [9] is employed. A detailed explanation on fuzzy complex FCTs comparison can be found in [2].

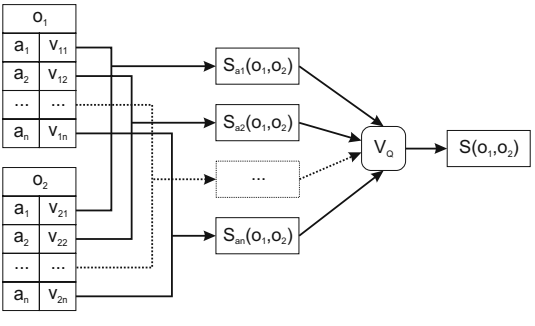


Fig. 2 Resemblance of two objects as proposed in CORM

3 ImmoSoftDataServerWeb (ISDSW)

ISDSW is a real estate search application based on the FORDBMS SDS. SDS supports flexible data handling using standard SQL syntax. The application allows

storage of real estates with flexible characteristics, and flexible querying over those characteristics.

ISDSW is implemented on the Java2EE platform, taking advantage of the features provided by Java on different application levels. The interface is built on JavaServer Pages and Servlet technology, and uses AJAX (Asynchronous JavaScript and XML). Its design allows the use of skins to personalize the look of the application, allowing its integration on existing web sites.

3.1 Features

Although flexible search is the main feature of the application, ISDSW provides a wide variety of services to users according to four different roles:

1. *Non Registered User* - Represents the initial state before register. This kind of user can define flexible queries using all the search capabilities provided by the application by means of web search forms. The user can obtain a registered account in order to obtain enhanced features.
2. *Registered User* - These users have a personal account that is used to bookmark interesting real estate profiles, store search criteria and post real estate offers. When new real estates are added to the system, they are checked against stored search conditions. If there is a match, an e-mail is sent to the user notifying the arrival of a new interesting offer.
3. *Sales Agent* - ISDSW provides sales agents with the necessary tools to manage a real estate customer base. Agents can manage the appointments requested by registered users to visit a particular real estate and can perform queries in the real estate database or in their customer bases. Additionally, new real states can be added by sales agents.
4. *Administrator* - Application administration tasks are done entirely via web. The administrator can manage user accounts, application elements and define geographic regions used in the geographic search.

3.2 Fuzzy Attributes in the Application

Real Estate brokerage is a business area where information is pervaded with imprecision. *Price*, *Area*, *Age*, etc... are attributes which are represented using an approximate value. In each case the flexible representation is useful for different reasons, sometimes it is due to ignorance of the exact value of the attribute and other times it is just an effect of the commercial side of the problem. In the following paragraphs data types used to model application attributes are discussed.

The number of *Rooms* or *Floors* is a well known value for owners and real estate agents, that is why we are representing it as a crisp value. Although the value is crisp, it can be queried flexibly because this is a particular case of trapezoidal distribution. Attribute *Price* is modelled as a *FuzzyNumber* type. Prices are subject to change depending on real estate conditions, market tendencies and even subjective factors

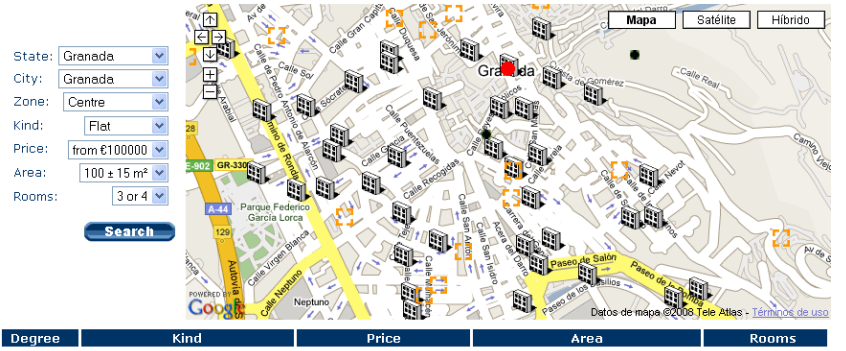
affecting the parties involved. Flexible representation of price ranges for real estates allows to incorporate additional semantics to the values. The trapezoidal representation used for fuzzy numbers makes possible to model a range of acceptable prices and a margin of negotiable prices. Attributes *Area* and *Age* are also represented as *FuzzyNumber* types. This case is different because the imprecision of the values is originated by ignorance of the exact value. Approximate values or intervals come in handy in these cases. In the case of *Kind* and *Conservation* the type used is Fuzzy Comparable Scalar, defining proximity relations for each value in the corresponding domain. Attribute *Location* uses a special *Geographic* type that is explained in detail in the following section.

3.3 Geographic Search

ISDSW search is enhanced with geographic features, users can define a geographical area where the real estates must be placed. Geographical conditions are applied in conjunction with flexible conditions obtaining the real estates fulfilling all the conditions in some degree.

The geographic interface is deployed using Google™ Maps API. The search form displays a map where users can navigate to define the area on which the real estate is going to be searched. It is also possible to define the search area using select lists, each list allows to select an state or area, a city and a street or zone. In Fig. 3 the geographic search interface with the select lists is depicted, along with results obtained for the query introduced.

The application manages geographic regions as a hierarchy with various granularity levels, as can be seen in Fig. 4. The most general regions are countries, and the most specific are street names with ZIP code or number. Each region has an identifier and a set of visualization coordinates. Every region references to the region where it belongs in the superior level, except for countries, because they are at the



Degree	Kind	Price	Area	Rooms
80%	Apartment	€120000	100 m²	3
73%	Flat	€155000	90 ± 15 m²	4
68%	Attic	€170000	80 ± 10 m²	4
29%	House	€166000	122 m²	2

Fig. 3 Query results obtained with Google Maps powered geographic search

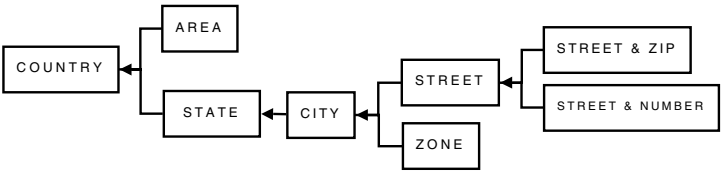


Fig. 4 Geographic levels included in the application

top level. Each geographic region has a self-describing name, except for area and zone. Area is used here as an aggregation of states, i.e. West Coast, Midwest, etc... while zone represents a polygonal shape over a city.

The map is synchronized with the selection lists, each time a selection is made, the map refreshes and shows the portion of the map corresponding to the visualization coordinates of the region selected. If the user prefers to navigate the map, search is performed according to the coordinates defined by the visualization window.

This geographical model allows to use certain degree of imprecision, real estates can be placed with precise coordinates or attached to one of the regions defined. Real estates of a particular region are selected as possible results if they fulfill the search conditions and if the intersection of the visualization window at query time with the coordinates of the region is not empty.

3.4 System Architecture

To implement the application we have chosen a layer based architecture in order to ease maintenance and update of components.

The application is divided in three layers that interact between them as can be seen in Fig.5.

1. *Presentation Layer:* Generates the user interface in order to provide an easy and comfortable interaction with the application. The main components of this layer are the Web Browser where the interface is visualized, JSP pages and JSP Tags that compose the interface, and the Web Server that processes JSP pages and sends resulting HTML to the user.
2. *Logic Layer:* The one that implements functionality and behavior expected from the application and provides methods to access that functionality. This layer receives data from the interface and processes it so that it can be used in the following layer. Components of this layer are Java Bean objects which enclose all application logic.
3. *Data Access Layer:* This layer allows application logic to access data stored in the FORDBMS. Data access is managed by data persistence objects, and it is performed using XML and FORDBMS fuzzy query syntax. This data persistence system is extensible and new persistence objects can be derived from it.

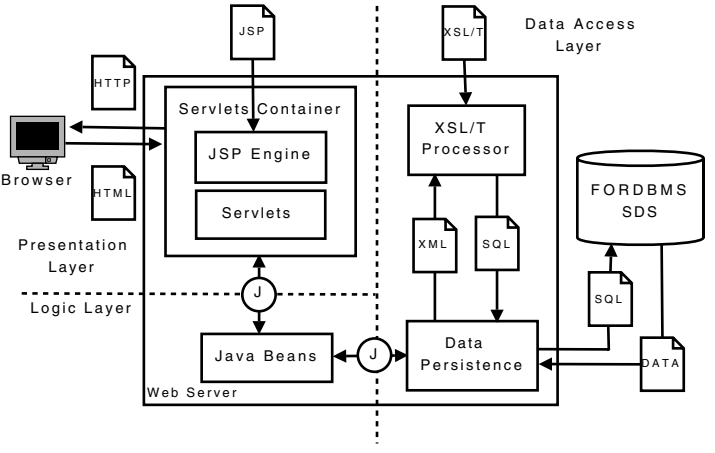


Fig. 5 General application architecture

Main components of this layer are the XSLT Parser, that translates user queries expressed in XML to fuzzy queries the FORDBMS can understand, and the FORDBMS itself (SDS), which executes fuzzy sentences and returns results.

The transformation from XML to fuzzy queries in SQL is performed to gain independence from particular query syntaxes providing reusability and adaptability.

Platform independence stands out as the main advantage of the proposed architecture. Other important features are its completely customizable *Interface*, which can be adapted easily using templates, its adaptable *Application Logic* that can be deployed to a wide variety of database schemas, and the XML based *Data Access Layer* which allows to gain independence from the underlying database syntax. These features provide a flexible framework adaptable to changes in the design of the interface, the database structure and the database language syntax.

The ISDSW system has been deployed for a local company as can be seen in [1]. This particular implementation of the system allows to give structure and provide storage for real estate classified advertisements using fuzzy attributes. The user can perform flexible searches using as query criteria price, type of real estate, geographic location and other real estate features.

4 Concluding Remarks and Future Work

In this paper we have presented ImmoSoftDataServerWeb, a real estate management application that takes advantage of fuzzy set theory to manage real estate offers. Fuzzy data management is possible thanks to Soft Data Server, a Fuzzy Object relational Database Management System, that is built as an extension of *Oracle*®.

Flexible search provides customers a wide range of results, similar to some degree to their requirements. This new set of extended results provides new business opportunities.

Future work will focus on the creation of an application framework based on SDS and ISDSW to generate custom fuzzy web search applications in a semi-automatic way. The layered design and the use of technologies to obtain platform independence makes the application fully customizable without too much effort.

Acknowledgements. This work has been partially supported by the Spanish "Ministerio de Ciencia y Tecnología" (MCYT) under grants TIN2006-07262/ and TIN-68084-C02-00, and the "Consejería de Innovación Ciencia y Empresa de Andalucía" (Spain) under research projects P06-TIC-01433, P06-TIC-01570 and P07-TIC-02611.

References

1. PuertaElvira.com. Real Estate Web Portal of Granada (2008), `http://www.puertaelvira.com/`
2. Barranco, C.D., Campaña, J.R., Medina, J.M.: Towards a fuzzy object-relational database model. In: Galindo, J. (ed.) Handbook of Research on Fuzzy Information Processing in Databases, ch. 17, pp. 431–461. Information Science Reference (2008)
3. Barranco, C.D., Campaña, J.R., Cubero, J.C., Medina, J.M.: A fuzzy object relational approach to flexible real estate trade. WSEAS Transactions on Information Science and Applications 2(2), 155–160 (2005)
4. Barranco, C.D., Campaña, J.R., Medina, J.M., Pons, O.: ImmoSoftWeb: a web based fuzzy application for real estate management. In: Favela, J., Menasalvas, E., Chávez, E. (eds.) AWIC 2004. LNCS (LNAI), vol. 3034, pp. 196–206. Springer, Heidelberg (2004)
5. Cubero, J.C., Marín, N., Medina, J.M., Pons, O., Vila, M.A.: Fuzzy object management in an object-relational framework. In: X Intl. Conf. of information processing and management of uncertainty in knowledge-based systems, pp. 1767–1774 (2004)
6. Galindo, J., Medina, J.M., Pons, O., Cubero, J.C.: A server for fuzzy SQL queries. In: Andreasen, T., Christiansen, H., Larsen, H.L. (eds.) FQAS 1998. LNCS (LNAI), vol. 1495, pp. 164–174. Springer, Heidelberg (1998)
7. ISO/IEC 9075:2003: Information Technology – Database languages – SQL. International Organization for Standardization (ISO), Geneva, Switzerland (2003)
8. ISO/IEC 9075-2:1999: Information Technology – Database Languages – SQL. International Organization for Standardization (ISO), Geneva, Switzerland (1999)
9. Marín, N., Medina, J.M., Pons, O., Sánchez, D., Vila, M.A.: Complex object comparison in a fuzzy context. Information and Software Technology 45, 431–444 (2003)
10. Medina, J.M., Pons, O., Vila, M.A.: GEFRED: A generalized model of fuzzy relational databases. Information Sciences 76(1-2), 87–109 (1994)

Proportional Load Balancing Using Scalable Object Grouping Based on Fuzzy Clustering*

Romeo Mark A. Mateo and Jaewan Lee

Abstract. In this paper, we present a scalable grouping of distributed objects based on the fuzzy clustering and proposes an intelligent object search with an optimal load sharing. The scalable grouping of objects contributes on identifying the appropriate server to forward the incoming request and efficiently balances the loads using the proposed equally proportional load distribution (EPLD). In the search process, the load balancing service uses the EPLD function to minimize the load variations where the multiple memberships of each object are used to distribute the loads in near equal proportions to the servers. Performance result showed that the proposed scalable grouping classifies more objects for availability of resources and minimizes the load variations within the servers.

1 Introduction

The performance of the network systems is improved by forwarding the process from heavily-loaded computer to the least-loaded computer. Previous researches in load sharing of networked computers propose techniques that dynamically allocate tasks within the network of computers. The implementation of the application system in distributed object environment provides a manageable load distribution of tasks [1, 2, 3]. The availability of services for transactions, concurrency control, security, events and persistent objects make it a desirable choice for use in many applications that are intended for use within an organization or a related of organizations [4]. There are various architectures to select on designing the interaction of distributed objects. CORBA introduced standards and techniques on designing distributed object platforms [5] where the standards were defined by the Object Management Group (OMG) [6]. Jini of Sun Microsystems is a distributed object implementation using Java objects which is based on the concept of federating groups. Most studies in distributed object architectures focus on efficient search of objects [1, 7, 8 ,9] and load balancing of objects [2, 10, 11]. Large implementation

Romeo Mark A. Mateo · Jaewan Lee

School of Electronic and Information Engineering, Kunsan National University 68 Miryong-dong, Kunsan, Chonbuk 573-701, South Korea

e-mail: {rmmateo, jwlee}@kunsan.ac.kr

* This research was financially supported by the Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation.

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 41–50.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

of distributed system is proposed in Globe project [12] where the state is encapsulated in the distributed shared objects and is use for distribution, consistency, and replication. Modular strategies overcome the complex interaction from objects, especially replication of objects, and led the researchers to propose object group models [13]. Object group models are designed to manage the system by grouping the objects and provide a singleton behavior of the grouped objects. The communications on object groups reflect the inter-dependence and take place from one group to another. To define, an object group is a set of objects related logically. A group acts as a logical addressable entity where an entity that requests a service from a group is a client of the group. Previous researches on object grouping method like in [3, 13] concentrate on the grouping mechanisms but lacks of knowledge based models which use the properties of objects as data.

Clustering is one of the popular data analysis methods and remarkably rich conceptual and algorithmic framework for data analysis and interpretation [14]. The structure revealed from clustering provides a good model for classification where each cluster forms a single class. In the research of cellular manufacturing by Ben-Arieh [15], presents an algorithm for grouping where the method uses a classical method of clustering and using agents to negotiate which group the object belongs. In [16], a clustering method of objects is used to enhance the trading service of CORBA. It shows a clustering scheme, centered on semantics, rather than schematics. The basis is to cluster services on their properties, which give service offers their semantics. Service offers which have similar properties are related by being clustered into one or more contexts. These classical methods are only concerned with the exact value to separate the data clusters and other useful information from the granules of data is obviously ignored.

This paper presents the scalable grouping of distributed objects based on fuzzy clustering and the intelligent object search based on the proposed equally proportional load distribution (EPLD) that are implemented by the services, namely, grouping, locator and load balancing services. The fuzzy clustering is used to provide a scalable access to resources. The grouping service handles the knowledge based grouping using fuzzy clustering to determine the fuzzy membership. The locator service processes the client request of objects in by collaborating with the deployed mobile agents to select the server based on the EPLD, implemented by the load balancing service, to minimize the load variation within the servers.

2 Background and Related Works

Cluster analysis divides data into groups such that similar data objects belong to the same cluster and dissimilar data objects to different clusters [14]. Partitioning methods construct c partition of data, where each partition represents a cluster and $c \leq$ n. There are several methods used to achieve the optimization of clustering. The most common one, named *c*-means [17] is a well-established way of clustering data. Partition matrices are utilized which are appealing to illustrate the structure of patterns. The objective function depends on the distances between vectors u_k and cluster centers c_i, and when the Euclidean distance is chosen as a distance function, the expression for the objective function is,

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \left(\sum_{k=1, \mathbf{u}_k \in C_i} m_{ik} \left\| \mathbf{u}_k - \mathbf{c}_i \right\|^2 \right) \tag{1}$$

The J is minimized by several iterations and stops if either the improvement over the previous iteration is below a certain tolerance or J is below a certain threshold value. However, the result only represents crisp membership and limits the scalability of classifying the object to another group. Also, there are possibility that clustering fails on classifying the data because of variation of compactness. In strong similarity to the hard clustering algorithms, fuzzy clustering is derived [17]. Fuzzy clustering is different from hard *c*-means, mainly because it employs fuzzy partitioning, where a point can belong to several clusters with degrees of membership. It does not try to assign each pattern to exactly one cluster, instead a degree of membership to each cluster is derived as well. In this paper, this approach is used to process the object properties in fuzzy clustering. The property-based clustering of objects in [16] uses a graphical hierarchy links to connect the object to the context or properties but did not consider processing the data values from the properties in data analysis.

Implementing load balancing to the distributed objects promotes QoS, scalability and dependability through the system. Solving the problem of optimal load sharing in object based systems is a necessary issue of distributed systems. Numerous middleware-based load balancing schemes are studied by researches like in the paper of Othman et. al. [17], the performance of round-robin and minimal dispersion load balancing schemes is analyzed. Using the round-robin provides a fast forwarding of request and has fast response time to clients request but the load distribution from the replica may not be equal. The disadvantage of the round-robin is solved by the minimal dispersion algorithm but it consumes time on determining the least loaded server where there is latency on the response time. Finally, the adaptive scheme is proposed where the two algorithms are used to implement the efficient load balancing. Fuzzy logic controller is used in load balancing the Jini services [2]. The approach works by using a fuzzy logic controller which informs a client object to use the most appropriate service such that load balancing among servers is achieved. Jini is used to simulate the middleware platform, on which the proposed approach and as well as other approaches are implemented and compared. Like in the research of Othman, the algorithm only considers a single type of object to process the task. In the discussion of our proposal, objects contain different properties based on the service it provides which makes the load distribution more complex. In the previous research [8], the adaptive load balancing of the distributed objects with different properties that forms object groups is considers. The coordination of the components to perform the adaptive load balancing provides an efficient load balancing. This proposal extends the issue of the nodes that have different type of objects and addresses the computational capacity of computers where the previous research [8] only uses the same type of objects in each server and each server assumes that it all servers host same object type.

3 Scalable Object Grouping Based on Fuzzy Clustering

The scalable object grouping uses fuzzy clustering and search method uses the proposed equally proportional load distribution. There are three services that implement the proposed algorithms which are the grouping, locator and load balancing services. Clustering of objects is managed by grouping service to re-organize the objects based on the properties. In [16], a clustering of object services is used to enhance the search of appropriate objects by the trading service in CORBA.

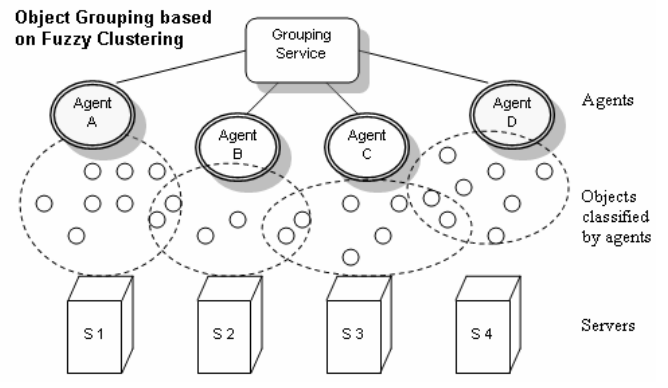


Fig. 1 Grouping of objects based on fuzzy clustering

In this paper, we use the fuzzy clustering to classify more objects in a group which promotes scalability of resources to the system. A server can host one or several objects and objects are grouped based on its properties. The fuzzy membership of an object is used whenever a request is classified to a group of objects. This study assumes that each server can contain different objects and after executing the fuzzy clustering, each object having an approximate or equal value of properties to another object is identify as a replica of that object. The approximation is measured by the fuzzy values from the rules extracted by the proposed algorithm. A mobile agent is assigned to contain the rules and information of the objects to be used in the search method. The objects are grouped by the grouping service and mobile agents are created and deployed in each host shown in Figure 1. After the procedure, each mobile agent updates its object information and fuzzy system. The objects overlap its membership to each mobile agent where some objects from agent *A* have a membership degree of belonging to agent *B*.

The fuzzy clustering algorithm starts on partitioning the collection of K data points specified by m-dimensional vectors k (k = 1, 2… K) into c fuzzy clusters, and finds a cluster center in each cluster, minimizing an objective function. The initialization of centers is critical to have the minimum iteration from the objective function. In this study the fuzzy sets are initialized by using the eigen decomposition method in Equation 2. Let E be a matrix of eigenvectors in a given square matrix eM and D be a diagonal matrix with the corresponding eigen values. As long as eM is a square matrix, eigen decomposition is processed in Equation 2.

$$eM = EDE^{-1} \tag{2}$$

After calculating the values of k in eigen decomposition, k is sorted from lowest to highest value based on eM. The values are arranged to a one dimensional array and prepare for calculating the fuzzy system for the fuzzy clustering. $d(A,B)$ is the function to determine the distance from set A to B and calculated by subtracting k_{max}, which is the maximum value of k and $k_{\min}$ is the minimum value of k divided by the number of clusters c. The overlapping of the fuzzy set is determined by o and added to the length of the set shown in Equation 3.

$$fuzzylength = d(A,B) = \frac{k_{\max} - k_{\min}}{c} + o \tag{3}$$

The length of the fuzzy set is used to calculate the initial value of the minimum ($Cmin_i$), maximum ($Cmax_i$) and center ($Ccen_i$) value of each fuzzy set. Initial values are presented by the following: $Cmin_i = k_{\min}$, $Cmax_i = k_{\min} + fuzzylength$, and $Ccen_i = k_{\min} + (fuzzylength / 2)$. In next iterations, the values of each fuzzy set are the addition of previous value j ($j=i-1$) and $fuzzylength / 2$. Also, every center of the group is initially set to $c_i = Ccen_i$. The objective function of fuzzy clustering is presented in Equation 4,

$$J(\mathrm{M}, c_1, c_2, ..., c_c) = \sum_{i=1}^{C} J_i = \sum_{i=1}^{C} \sum_{n=1}^{N} m_{in}^q d_{in}^2, \tag{4}$$

where the membership function (m^q_{in}) maps the fuzzy membership values from the fuzzy system initialized from Equation 4. Every time an update occurs in c_i then the center of the fuzzy sets in the fuzzy system also adjusts where, $Ccen_i = c_i$. These equations are used to determine the center of point of the cluster and center from the fuzzy system. After the generation of the fuzzy clustering structure, it is used to classify each object shown in Equation 5.

$$objectClass_i(x_k) = \begin{cases} 1 & if\ m_{ik}(u_i) > \Phi, \\ 0 & otherwise. \end{cases} \tag{5}$$

A threshold value controls the membership of an object presented by Φ and $\Phi < 1$. The value depends on the domain of the system which needs more objects to classify. Equation 5 is the function to classify the objects.

4 Load Distribution Based on Equal Proportions

Most of the load balancing schemes for distributed objects uses an adaptive scheme [8, 2, 6] but do not tackle more on the optimal minimization of the load variation within the servers, especially considering different load capacities of computers to perform the task. This paper defines a load as the current object in process and the capacity of a server refers to the number of objects hosting the server. We assumed that the processing power of a server depends on the number of object processing requests. The more request processing in a server, the more it

uses its processing capacity. We propose the equally proportional load distribution (EPLD) which approximately distributes equally the loads based on the current loads and number of objects hosting by the server. The algorithm tries to proportionate the number of idle objects and current loads to all servers. Whenever a request is occurred then the locator service determines which class it belongs and finds the appropriate object. After determining the class, it chooses the server that will bind its object to the request based on Equation 5. The selected servers (*SS*) are collected to *N* (*SS*=*N*) and process the incoming load in the proposed equally proportional load distribution. First, the procedure determines the mean load (μ_l) of each state which is the average of p_n, a ratio of currently accessed objects (*l*) and total objects in a server (*T*) in Equation 7.

$$\mu_i = \frac{1}{N}\sum_{n=1}^{N} p_n \tag{6}$$

$$p_n = \frac{l_n + f(x)}{T_n} \tag{7}$$

In calculating the mean in Equation 6, the incoming load is added to the selected *n* server represented by the function *f*(x) in Equation 7. The *f*(*x*) forwards the load to a server that has an index of *n*, else returns zero, and then the value of *x* is used in Equation 8. In every state in *S*, this procedure is done alternately to a server where *n*=*i*.

$$f(x) = \begin{cases} x, \ if \ n = i \\ 0, \ else \end{cases} \tag{8}$$

$$\sigma_i = \frac{1}{N}\sum_{n=1}^{N} \left| p_n - \mu_i \right|^2 \tag{9}$$

In Equation 9, represents a single state where it calculates the load variance all server. The smaller load variance means that the loads are well distributed and make sure that all servers are contributing in processing the task. The selection of minimum load variance is expressed in Equation 10.

$$CandidateServer = getindex(\min\{\sigma_1, \sigma_2, ..., \sigma_i\}) \tag{10}$$

The candidate server selection in Equation 10 is the final procedure in the algorithm which chooses the smallest load variance in *S*. After determining the candidate server, the procedure selects the object from the server. The agent forwards the object ID of the selected object to the client. If the selected object is currently accessed by clients, then the server chooses another object which is idle or less busy than the object currently chosen to forward the request.

5 Experimental Evaluation

Each object from the object groups are grouped by using the proposed grouping service. After the procedure, grouping service creates mobile agents and assigns

the fuzzy values and object information to mobile agents. The fuzzy membership of each objects are assigned to a group based on its properties. The object grouping schemes were evaluated by adding all objects members in each group. Equation 11 shows the calculation of object count in a cluster where ObjectCount_i is incremented to 1 if the membership of the object is classified to class *i*. The crisp value of *k*-means is easily determined by comparing the Euclidean distance of the object properties to each cluster centers represented by $d_i=|c_i - x|$. The class *i* that have the smallest d_i is chosen group and ObjectCount_i incremented. In the proposed scalable grouping, the objects can also be classified to other classes. Unlike in *k*-means, a crisp separation from the clusters limits the scalability of objects to be classified to other classes. Using the proposed scalable grouping, ObjectCount_i is incremented to 1, if $m_i(x_k) > \Phi$ and an object can belong to another class.

$$ObjectCount_i = \sum_{k=1}^{K} m_i(x_k) \tag{11}$$

We determine the performance of the proposed EPLD to other previous works [3, 10] by calculating the variance of the load distribution in Equation 12 where *X* is the proportion of the current load and number of objects of the servers and *M* is the mean load proportions. In the other works like in [3] discuss on considering the load variance but not having the minimization of load variance. The study only used a certain threshold to compare to the load variance and switch the mechanism whereas our proposed load distribution tackles the issue of the minimal load variation.

$$S^2 = \frac{1}{N}\sum_{n=1}^{N}\left|X - M\right|^2 \tag{12}$$

5.1 Simulation Result

In our simulation, a synthetic data was used to compare the performance of the proposed scalable grouping and other methods. A standard distribution of generating random data, which contains 5 attributes and 150 tuples were done, and then make these data as object properties. To perform the online simulation of the proposed algorithm, the properties of each object contained pattern from the synthetic data. All objects were classified through the grouping procedure based on the properties. Simulation result from Equation 11 is shown in Table 1. In the proposed scalable grouping, the object can have several memberships depending on the threshold (Φ) specified while in *k*-means, objects were grouped only in one class which has a crisp membership. Setting up a small threshold means that more objects are classified. Fuzzy clustering provides more objects and these objects are used as object replicas for providing scalability of the system while in *k*-means, the objects are classified only in a single group and cannot share its functionality to other groups.

The scalability of objects using the function is used to distribute the loads proportionally within the server. We used three servers with the following number of objects; A=50, B=62 and C=38. The objects generated were distributed to the

Table 1 Membership count of objects in each group using fuzzy clustering with Φ (0.05, 0.1, 0.15, 0.2) and *k*-means

Groups	neuro-fuzzy clustering				*k*-means
	Φ_1=0.05	Φ_2 =0.1	Φ_3 =0.15	Φ_4 =0.2	
Class 1	62	59	59	54	50
Class 2	75	74	70	69	62
Class 3	77	75	72	71	38
Total	214	208	201	194	150

servers and process the scalable grouping. After the grouping, the accessing of objects by clients was simulated and the load balancing service executes the EPLD to distribute the loads. The result of simulation using Equation 12 is shown in Table 2 where the variance of the current load and idle objects is calculated. The result of using the EPLD provides a smaller load variation through the servers with a value of 0.00875 on current load compared to classical minimal dispersion or least loaded server scheme with a value of 0.1378 on current load.

Table 2 Variation of load distribution within the servers using EPLD and LL

C: A=50; B=62; 38, l: A=50; B=30; C=20				
Server			EPLD	LL
	Current loads	Idle objects	Current loads	Idle objects
A	33	17	34	16
B	42	20	33	29
C	25	13	33	5
σ	0.00875	0.00875	0.1378	0.1376

The result of from Table 1 provide a scalable classification of objects while in Table 2 shows that the scalable grouping contributes on the minimization of the load variance within the system .

6 Conclusions and Future Work

The integration of the knowledge-based model to the system contributes on providing accurate services and acquires system optimal performance. In this paper, a scalable grouping for distributed object based on fuzzy clustering and optimal load distribution based on the proposed EPLD are presented. The grouping service handles the object grouping based on fuzzy clustering which determines the fuzzy membership of each object and the locator service implements the intelligent search for appropriate object. The fuzzy clustering uses the fuzzy system which is initialized by sorting the data based on eigen decomposition and creating the fuzzy set and fuzzy rules. In the goal of providing the minimal load variation, the EPLD which is implemented by the load balancing service is proposed. Before the client

accesses the appropriate object, the load balancing service executes the EPLD which selects the server to forward the request which have the minimal load variation. Simulation showed more object classified using the proposed scalable grouping in each group. The scalable grouping using different thresholds was compared to *k*-means and shows that an averaging of 36% more objects can be used by the system. The load variance was also simulated from EPLD which is significantly smaller (0.00875) compared to minimal dispersion scheme or least load scheme (0.1378). The future work is to implement the knowledge-based object grouping to a specific application to generate real data and adjust the algorithm for desired results.

References

1. Damiani, E.: A fuzzy stateless approach to load distribution for object-oriented distributed environments. International Journal of Knowledge-Based Intelligent Engineering System 3(4), 240–253 (1999)
2. Kwok, Y.K., Cheung, L.S.: A new fuzzy-decision based load balancing system for distributed object computing. Journal of Parallel and Distributed Computing 2(64), 238–253 (2004)
3. Mateo, R.M.A., Yoon, I., Lee, J.: Cooperation model for object group using load balancing. International Journal of Computer Science and Network Security 6(12), 138–147
4. Coulouris, G., Dollimore, J., Kindberg, T.: Distributed systems: concepts and design, 4th edn. Addison-Wesley, Reading (2005)
5. Yang, Z., Duddy, K.: CORBA: A platform for distributed object computing. ACM Operating Systems Review 30(2), 4–31 (1996)
6. Object Management Group, `http://www.omg.org`
7. Badidi, E., Keller, R.K., Kropf, P.G., Van Dongen, V.: The design of a trader-based CORBA load sharing service. In: Proceedings of the 12th International Conference on Parallel and Distributed Computing Systems, pp. 75–80 (1999)
8. Baggio, A., Ballintijn, G., Van Steen, M., Tanenbaum, A.S.: Efficient tracking of mobile objects in globe. The Computer Journal 44(5), 340–353 (2001)
9. Van Steen, M., Ballintijn, G.: Achieving scalability in hierarchical location services. In: Proceedings of the 26th International Computer Software and Applications Conference (2002)
10. Othman, O., O'Ryan, C., Schmidt, D.C.: The design of an adaptive CORBA load balancing service. IEEE Distributed Systems Online 2(4) (2001)
11. Schnekenburger, T.: Load balancing in CORBA: A survey of concepts, patterns, and techniques. The Journal of Supercomputing 15, 141–161 (2000)
12. Homburg, P., Van Steen, M., Tanenbaum, A.S.: An architecture for a wide area distributed system. In: Proceedings of the 7th ACM SIGOPS European Workshop, pp. 75–82 (1996)
13. Felber, P., Guerraoui, R.: Programming with object groups in CORBA. IEEE Concurrency 8(1), 48–58 (2000)
14. Anderberg, M.R.: Cluster analysis for applications. Academic Press, New York

15. Ben-Arieh, D., Sreenivasan, R.: Information analysis in a distributed dynamic group technology method. International Journal of Production Economics 60-61(1), 427–432 (1999)
16. Craske, G., Tari, Z.: A property-based clustering approach for the CORBA trading service. In: Proceeding of Int'l Conference on Distributed Computing Systems, pp. 517–525 (1999)
17. Bezdek, J.C.: Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York (1981)

Multilevel Image Segmentation Using OptiMUSIG Activation Function with Fixed and Variable Thresholding: A Comparative Study

Sourav De, Siddhartha Bhattacharyya, and Paramartha Dutta

Abstract. An optimized multilevel sigmoidal (OptiMUSIG) activation function for segmentation of multilevel images is presented. The OptiMUSIG activation function is generated from the optimized class boundaries of input images. Results of application of the function with fixed and variable thresholding mechanisms are demonstrated on two real life images. The proposed OptiMUSIG activation function is found to outperform the conventional MUSIG activation function using both fixed and variable thresholds.

1 Introduction

Image segmentation involves classification and clustering of image data based on shape, color, position, texture and homogeneity of image regions. It finds applications in satellite image processing, surveillance and astronomical applications.

Several classical methods of image segmentation are reported in the literature [5, 9, 20]. Guo *et al.* [10] proposed an unsupervised stochastic model based segmentation approach. In this method, relevant parameter estimation is made on the basis of Bayesian learning. Subsequently, a competitive power-value based approach is used to segment the images into different classes. Malik *et al.* [14] treated image segmentation as a graph partitioning problem. A graph theoretic framework

Sourav De
Department of Computer Science and Information Technology, University Institute of Technology, The University of Burdwan, Burdwan - 713 104
e-mail: sourav.de79@gmail.com

Siddhartha Bhattacharyya
Department of Computer Science and Information Technology, University Institute of Technology, The University of Burdwan, Burdwan - 713 104
e-mail: siddhartha.bhattacharyya@gmail.com

Paramartha Dutta
Department of Computer and System Sciences, Visva-Bharati, Santiniketan - 721 325
e-mail: paramartha.dutta@gmail.com

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 53–62.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

of normalized cuts is used to divide the image into regions with coherent texture and brightness. A pyramidal image segmentation technique with the fuzzy c-means clustering algorithm is reported in [16]. In this method, a root labeling technique is applied to divide each layer of a pyramid into a number of regions for merging the regions of each layer with the highest resolution. The minimum number of regions is automatically determined by a cluster validity function.

Neural networks [18] have been applied for clustering of similar data by selection and assignment of underlying prototypes to a pattern based on its distance from the prototype and the data distribution. In this approach, the number of clusters are determined automatically. Segmentation by a self-organizing neural network (SONN) is presented in [13]. The proposed approach is based on the orientation of textures within the images. Multi modal image segmentation has been carried out by an improved Hopfield's neural network [17]. Kohonen's self-organizing feature map (SOFM) [12] is a competitive neural network which enables preservation of important topological information through an unsupervised learning process. SOFM has been used for pixel-based segmentation of images [1]. In this method, each pixel is initially assigned to a scaled family of differential geometrical invariant features. Input pixels are then clustered into different regions using SOFM. Alirezaie *et al.* [2] proposed an unsupervised segmentation technique for magnetic resonance images (MRI) using SOFM. The multilayer self organizing neural network (MLSONN) [7] is a feedforward architecture, suitable for the extraction of binary objects from noisy and blurred image scenes. MLSONN works in a self supervised manner based on pixel neighborhood information. Since this network uses the generalized bilevel sigmoidal activation function, it is incapable of segmenting multilevel images.

Bhattacharyya *et al.* [6] introduced a multilevel sigmoidal (MUSIG) activation function for effecting multilevel image segmentation by an MLSONN architecture. The different transition levels of the MUSIG activation function are determined by the number of gray scale objects and the representative gray scale intensity levels. However, the function resorts to a single fixed point thresholding mechanism assuming equal and homogeneous responses from all the representative levels of gray. In reality however, images exhibit a varied amount of heterogeneity.

Genetic algorithms (GAs) [8] have been resorted to for achieving optimized image segmentation solutions. In [4], a set of hyperplanes is generated to classify patterns based on genetic algorithm. Mean square error (MSE) estimation techniques have been used to segment an image into regions [11]. The optimized error estimation criterion is dependent on the shape and location of the underlying regions. Yu *et al.* [19] introduced an image segmentation method using GA combined with morphological operations. In this method, morphological operations are used to generate the new generations of GA. A score of GA based image segmentation approaches are available in [3, 15].

In this article, genetic algorithm is applied to generate the optimized class boundaries of a multilevel sigmoidal activation function to be used for segmentation of gray scale images into different classes. The proposed optimized multilevel sigmoidal activation (OptiMUSIG) function is generated by these dynamically generated class boundaries with fixed threshold. In addition, a variable thresholding

mechanism is also incorporated in the OptiMUSIG function through these dynamically generated class boundaries. The multilevel images are segmented into multiple scales of gray by using a single MLSONN architecture, characterized by the designed fixed and variable threshold based OptiMUSIG activation function. A comparative study of the thresholding mechanisms with the heuristically generated multilevel sigmoidal activation function and the proposed optimized counterpart, is illustrated using two real life multilevel images. The standard correlation coefficient between the original and the segmented images is used as a figure of merit. Results show that the OptiMUSIG function outperforms the conventional MUSIG activation function for both fixed and variable thresholding mechanisms.

2 Multilayer Self-Organizing Neural Network (MLSONN)

The multilayer self organizing neural network (MLSONN) [7] is a feedforward network architecture characterized by a neighborhood topology-based network interconnectivity. It consists of an input layer, any number of hidden layers and an output layer. The network operates in a self supervised mode featuring backpropagation of errors. The output layer neurons are connected with the input layer neurons on a one-to-one basis. The system errors are calculated from the linear indices of fuzziness [7] in the network outputs obtained. The processing neurons of the MLSONN architecture are activated by the standard bilevel sigmoidal activation function, given by

$$y = f(x) = \frac{1}{1 + e^{-\lambda(x-\theta)}} \tag{1}$$

where, λ decides the slope of the function and θ is a fixed threshold/bias value. A detailed analysis of the architecture and operation of the MLSONN architecture can be found in [7].

3 Optimized Multilevel Sigmoidal (OptiMUSIG) Activation Function

The MLSONN architecture characterized by the standard sigmoidal activation function, is able to map the input image into two levels, one darkest (0) level and one brightest (1) level of gray. In order to segment multilevel images, the MLSONN architecture resorts to a multilevel extension (MUSIG) of the bilevel sigmoidal activation function [6]. The MUSIG activation function is given by [6]

$$f_{\text{MUSIG}}(x; \alpha_\beta \ c_\beta) = \sum_{\beta=1}^{K-1} \frac{1}{\alpha_\beta + e^{-\lambda[x-(\beta-1)c_\beta-\theta]}} \tag{2}$$

where, α_β represents the multilevel class responses and is denoted as

$$\alpha_\beta = \frac{C_N}{c_\beta - c_{\beta-1}} \tag{3}$$

Here, β represents the gray scale object index $(1 \leq \beta \quad K)$ and K is the number of classes in the segmented image. The α_β parameter determines the number of transition levels/lobes in the resulting MUSIG function. C_N is the maximum fuzzy membership of the gray scale contribution of pixel neighborhood geometry. c_β and $c_{\beta-1}$ represent the gray scale contributions of the β^{th} and $(\beta-1)^{th}$ classes, respectively. The parameter(θ), used in the MUSIG activation function, is fixed and uniform.

The class boundaries used by the MUSIG activation function are selected heuristically from the gray scale histogram of the input images, assuming the homogeneity of the underlying image information. This means that the image context information does not get reflected in the segmentation procedure. However, in real life images, the image information is heterogeneous in nature, thereby implying that the class boundaries (as well as the class responses) would differ from one image to another. This heterogeneity can be incorporated into the MUSIG activation function by generating optimized class boundaries from the image context.

An optimized form of the MUSIG activation function, applying such optimized class boundaries, can be represented as

$$f_{\text{OptiMUSIG}}(x; \alpha_{\beta_{\text{opt}}} \; c_{\beta_{\text{opt}}}) = \sum_{\beta=1}^{K-1} \frac{1}{\alpha_{\beta_{\text{opt}}} + e^{-\lambda[x-(\beta-1)c_{\beta_{\text{opt}}}-\theta]}} \tag{4}$$

where, $c_{\beta_{\text{opt}}}$ are the optimized gray scale contributions corresponding to optimized class boundaries. $\alpha_{\beta_{\text{opt}}}$ are the respective optimized multilevel class responses. In the expression of the OptiMUSIG activation function given in Eq. 4, the threshold value (θ) is fixed and is not dependent on the class boundaries. A variable threshold parameter, $\theta_{\text{var}_{\text{opt}}}$ depending on the optimized class boundaries can be incorporated into the OptiMUSIG activation function as

$$f_{\text{OptiMUSIG}}(x; \alpha_{\beta_{\text{opt}}} \; c_{\beta_{\text{opt}}}) = \sum_{\beta=1}^{K-1} \frac{1}{\alpha_{\beta_{\text{opt}}} + e^{-\lambda[x-(\beta-1)c_{\beta_{\text{opt}}}-\theta_{\text{var}_{\text{opt}}}]}} \tag{5}$$

where, $\theta_{\text{var}_{\text{opt}}}$ is represented as

$$\theta_{\text{var}_{\text{opt}}} = c_{\beta_{\text{opt}}-1} + \frac{c_{\beta_{\text{opt}}} - c_{\beta_{\text{opt}}-1}}{2} \tag{6}$$

Similarly, a variable threshold θ_{var} can be computed from the heuristic class boundaries for the conventional MUSIG activation function given in Eq. 2, which is given as

$$\theta_{\text{var}} = c_{\beta-1} + \frac{c_\beta - c_{\beta-1}}{2} \tag{7}$$

An OptiMUSIG activation function generated for $K = 8$ with optimized class responses and fixed threshold is shown in Fig. 1(a). Fig. 1(b) shows the designed variable threshold based OptiMUSIG activation function for $K = 8$.

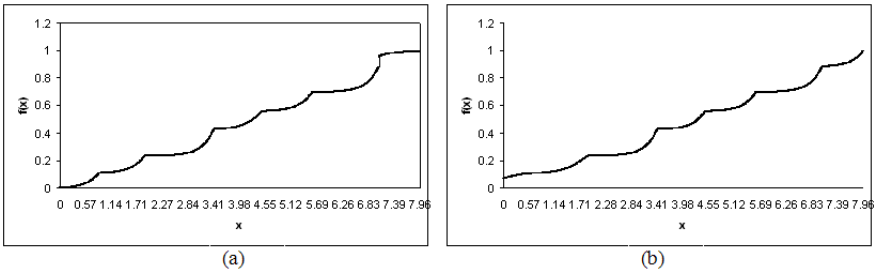


Fig. 1 OptiMUSIG activation function using optimized class responses (a) with fixed threshold (b) with variable threshold

Since the MLSONN architecture operates in a self-supervised manner, the system errors are computed using the subnormal linear indices of fuzziness [6].

4 Principle of MLSONN Based Optimized Multilevel Image Segmentation

The objective of optimized multilevel image segmentation has been achieved in the following three phases.

1. **Optimized class generation:** This phase marks the generation of the optimized class boundaries for a particular image. The number of classes (K) and the pixel intensity levels are used in this GA based optimized procedure characterized by a single point crossover operation.
2. **OptiMUSIG function generation:** This phase marks the generation of the optimized form of the MUSIG activation function. The requisite $\alpha_{\beta_{\text{opt}}}$ parameters are derived from the corresponding optimized $c_{\beta_{\text{opt}}}$ parameters (obtained in the previous phase) using Eq. 3. These $\alpha_{\beta_{\text{opt}}}$ parameters are further employed to obtain the different transition levels of the OptiMUSIG activation function.
3. **MLSONN based multilevel image segmentation:** In this final phase, real life multilevel images are segmented using an MLSONN architecture guided by the designed OptiMUSIG activation function with fixed and variable thresholds.

The entire procedure of multilevel image segmentation can be best illustrated by the following algorithm.

```
1 Begin
```

Optimized class generation phase

```
2 iter:=0
3 Initialize Pop[iter]
```

Remark: **Pop[iter]** is the initial population of class boundaries **cbounds**.

```
4 Compute F(Pop[iter])
```

Remark: **F** is the fitness function (correlation).

```
5 Do
6 iter:=iter+1
7 Select Pop[iter]
```

Remark: Selection of better fit chromosomes.

```
8 Crossover Pop[iter]
```

Remark: GA crossover operation.

```
9 Mutate Pop[iter]
```

Remark: GA mutation operation.

```
10 Loop Until (F(Pop[iter])-F(pop[iter-1])<=eps)
```

Remark: eps is the tolerable error.

OptiMUSIG function generation phase

```
11 Generate OptiMUSIG
```

Remark: Optimized MUSIG activation is formed with optimized **cbounds**.

MLSONN based Multilevel image segmentation phase

```
12 Segment input image with OptiMUSIG and MLSONN
```

Remark: MLSONN architecture is applied to segment input images using the Opti-MUSIG activation function.

```
13 End
```

5 Results

Applications of the proposed approach using the OptiMUSIG activation with fixed and variable thresholding mechanisms, have been demonstrated on an 8-class segmentation of multilevel Lena and Baboon images of dimensions 256×256. A value of $\theta = 2$ has been used for the fixed thresholding process. A value of $\lambda = 4$ is

Table 1 Optimized class boundaries for two multilevel images

	Lena				Baboon			
Level	Set 1	Set 2	Set 3	Set 4	Set 1	Set 2	Set 3	Set 4
1	0	0	0	0	0	0	0	0
2	28	36	38	35	40	68	73	62
3	60	69	70	68	78	91	110	88
4	109	101	107	110	114	112	133	121
5	143	127	129	150	140	141	153	139
6	179	152	152	177	164	169	177	164
7	226	191	199	216	191	190	192	196
8	255	255	255	255	255	255	255	255

used for the slope parameter. The four sets of optimized class boundaries derived from the genetic algorithm based optimization procedure, are shown in Table 1. The segmented multilevel images obtained by the variable and fixed threshold based OptiMUSIG activation function with optimized class responses pertaining to the first set of Table 1, are shown in Fig. 2.

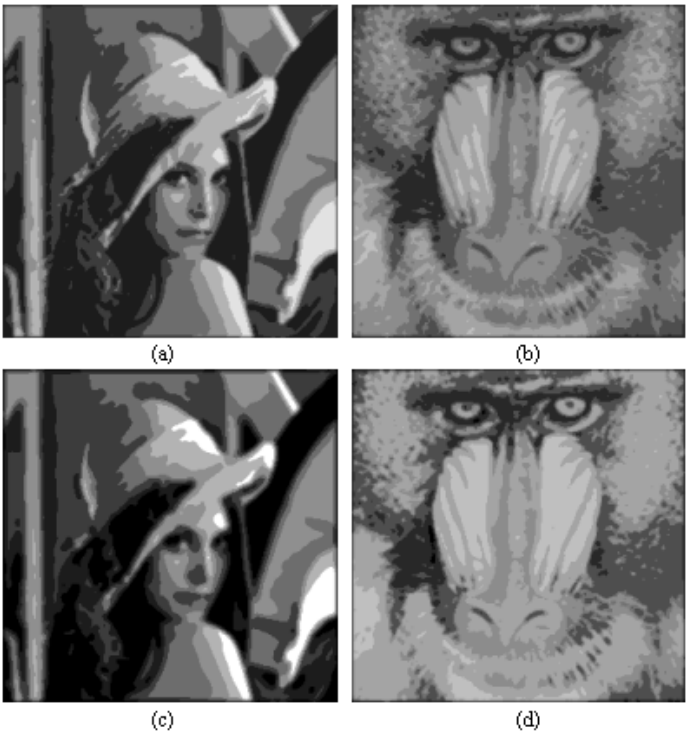


Fig. 2 8-class segmented test images using OptiMUSIG activation function (a)(b) with variable threshold (c)(d) with fixed threshold

Table 2 Fixed class boundaries of two multilevel images and the corresponding ρ

Image	Threshold	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	ρ
Lena	θ_{var}	0	30	60	85	160	190	223	255	0.9042
	θ	0	28	56	110	140	180	225	255	0.8889
Baboon	θ_{var}	0	33	67	95	124	182	220	255	0.8189
	θ	0	25	95	120	140	175	215	255	0.8033

The performance of the OptiMUSIG activation function has been further compared with the standard MUSIG activation function with fixed and variable thresholding mechanisms for the self-sufficiency of the treatment. The fixed class boundaries $\{L_1\ L_2\ L_3\ L_3\ L_4\ L_5\ L_6\ L_7\ L_8\}$ applied for the MUSIG activation function, are shown in the Table 2. The corresponding segmented multilevel images obtained by the MUSIG activation function with variable and fixed thresholding mechanisms are shown in Fig. 3.

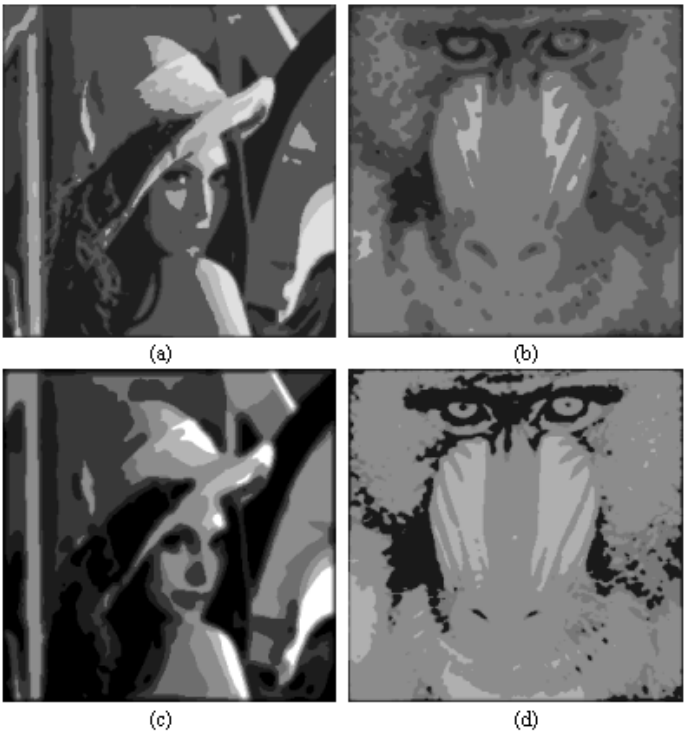


Fig. 3 8-class segmented test images using fixed threshold based MUSIG activation function with fixed class responses (a) Lena image (b) Baboon image

5.1 Performance Evaluation of OptiMUSIG Activation Function

The standard measures of correlation (ρ) between the original and segmented multilevel images have been used as the figure of merit for the segmentation process. The ρ values obtained by variable and fixed threshold based OptiMUSIG guided segmentation for different sets of optimized class boundaries, are shown in Table 3. The corresponding ρ values for MUSIG based segmentation with variable and fixed thresholding mechanisms are shown in Table 2 (9^{th} column).

It is evident from Tables 2 and 3 that the OptiMUSIG activation function outperforms its conventional counterpart irrespective of the thresholding mechanism employed.

Table 3 Correlation coefficients (ρ) using OptiMUSIG activation function with variable and fixed thresholding

Threshold	Lena				Baboon			
	Set 1	Set 2	Set 3	Set 4	Set 1	Set 2	Set 3	Set 4
$\theta_{\text{var}_{\text{opt}}}$	0.9500	0.9602	0.9545	0.9466	0.9358	0.9239	0.9314	0.9206
θ	0.9311	0.9008	0.9265	0.9347	0.9144	0.9113	0.9102	0.8841

6 Discussions and Conclusion

A novel approach for multilevel image segmentation using an MLSONN architecture guided by an optimized MUSIG (OptiMUSIG) activation function with fixed and variable threshold values, is presented in this article. The OptiMUSIG activation function is designed based on optimized class boundaries of input multilevel images. Better segmentation is achieved by the proposed activation as compared with a heuristically designed one. The authors are currently engaged in applying the proposed approach for segmentation of true color images.

Acknowledgements. The authors would like to acknowledge University Institute of Technology, The University of Burdwan, Burdwan and Visva-Bharati, Santiniketan for the infrastructures and logistic supports provided for carrying out this work.

References

1. Ahmed, M.N., Farag, A.A.: Two-stage neural network for volume segmentation of medical images. Pattern Recognition Letters 18(11-13), 1143–1151 (1997)
2. Alirezaie, J., Jernigan, M.E., Nahmias, C.: Automatic Segmentation of Cerebral MR Images using Artificial Neural Networks. IEEE Transactions on Nuclear Science 45(4), 2174–2182 (1998)

3. Al-Muhairi, H., Fleury, M., Clark, A.F.: Computationally Efficient Quantitative Testing of Image Segmentation with a Genetic Algorithm. In: Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, pp. 783–790 (2007)
4. Bandyopadhyay, S., Murthy, C.A., Pal, S.K.: Pattern Classification with genetic algorithms. Pattern Recognition Letters 16, 801–808 (1995)
5. Banham, M.R., Katsaggelos, A.K.: Digital Image Restoration. IEEE Signal Processing Magazine 14(2), 24–41 (1997)
6. Bhattacharyya, S., Dutta, P., Maulik, U.: Self Organizing Neural Network (SONN) based Gray Scale Object Extractor with a Multilevel Sigmoidal (MUSIG) Activation Function. Journal of Foundations of Computing and Decision Sciences 33(2) (2008)
7. Ghosh, A., Pal, N.R., Pal, S.K.: Self-Organization for Object Extraction Using a Multilayer Neural Network and Fuzziness Measures. IEEE Transactions on Fuzzy Systems 1(1), 54–68 (1993)
8. Goldberg, D.E.: Genetic Algorithm in Search Optimization and Machine Learning. Addison-Wesley, New York (1989)
9. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice Hall, Englewood Cliffs (2002)
10. Guo, G., Ma, S.: Bayesian Learning, Global Competition, and Unsupervised Image Segmentation. In: Proceedings of Fourth International Conference on Signal Processing, pp. 986–989 (1998)
11. Haseyama, M., Kumagai, M., Kitajima, H.: A Genetic Algorithm based Image Segmentation for Image Analysis. Proceedings of IEEE, 3445–3448 (1999)
12. Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (1995)
13. Lee, W., Kim, W.: Self-Organization Neural Network for multiple texture image segmentation. Proceedings of IEEE Tencon, 730–733 (1999)
14. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and Texture Analysis for Image Segmentation. International Journal of Computer Vision 43(1), 7–27 (2001)
15. Paulinas, M., Uinskas, A.: A Survey of Genetic Algorithms Applications for Image Enhancement and Segmentation. Proceedings of Information Technology and Control 36(3), 278–284 (2007)
16. Rezaee, M.R., Zwet, P.M.J., Lelieveldt, B.P.F., Geest, R.J., Reiber, J.H.C.: A Multiresolution Image Segmentation Technique Based on Pyramidal Segmentation and Fuzzy Clustering. IEEE Transactions on Image Processing 9(7), 1238–1248 (2000)
17. Rout, S., Srivastava, S.P., Majumdar, J.: Multi-modal image segmentation using a modified Hopfield neural network. Pattern Recognition 31(6), 743–750 (1998)
18. Vinod, V.V., Chaudhury, S., Mukherjee, J., Ghose, S.: A Connectionist Approach for Clustering with Applications in Image Analysis. IEEE Transactions on Systems, Man and Cybernetics 24(1), 365–384 (1994)
19. Yu, M., Eua-anant, N., Saudagar, A., Udpa, L.: Genetic Algorithm Approach to Image Segmentation Using Morphological Operations. Proceedings of IEEE, 775–779 (1998)
20. Zhengmao, Y., Mohamadian, H., Yongmao, Y.: Gray level image processing using contrast enhancement and watershed segmentation with quantitative evaluation. In: International Workshop on Content-Based Multimedia Indexing, pp. 470–475 (2008)

Artificial Neural Networks Modeling to Reduce Industrial Air Pollution

Zvi Boger

Abstract. Nitric acid production plants emit small amounts of nitrogen oxides (NOx) to the environment. As the regulatory authorities demand the reduction of the resulting air pollution, existing plants are looking for economical ways to comply with this demand. Several Artificial Neural Networks (ANN) models were trained from several months of operating plant data to predict the NOx concentration in the tail gas, and their total amount emitted the environment. The training of the ANN model was done by the Guterman-Boger algorithm set that generates a non-random initial connection weights, suggests a small number of hidden neurons, avoids, and escapes from, local minima encountered during the training. The ANN models gave small errors, 0.6% relative error on the NOx concentration prediction and 0.006 kg/hour on daily emission in the 20-45 kg NOx/hour range. Knowledge extraction from the trained ANN models revealed the underlying relationships between the plant operating variables and the NOx emission rate, especially the beneficial effect of cooling the absorbed gas and reticulating liquids in the absorption towers. Clustering the data by the patterns of the hidden neurons outputs of auto-associative ANN models of the same data revealed interesting insights.

1 Introduction

Nitric acid production plants emit small amounts of nitrogen oxides (NOx) to the environment. As the regulatory authorities demand the reduction of the resulting air pollution, existing plants are looking for economical ways to comply with this demand. One way is to find out if there is a potential to optimize the current operating policies, by creating a model of the plant operation relationship with the NOx emmission. The suggested use of artificial neural networks (ANN) modeling techniques in industrial plants, in which the model is learned from data of the plant behavior, often arouses strong emotions. ”No complicated equations! No

Zvi Boger
OPTIMAL – Industrial Neural Systems Ltd., Be’er Sheva 84243 Israel
Optimal Neural Informatics LLC, Pikesville, MD, 21208, USA
e-mail: optimal@peeron.com

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 63–71.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

man-years of development effort!" cheer the proponents. "No detailed equations? No reliability!" counter the opponents.

Even so, the use of ANN modeling in industrial plants is spreading, as other modeling methods are costly, both in resources and time, to fully meets the requirements of fault diagnosis or plant operation optimization.

"Soft sensor" is the accepted name for ANN model (or other model) that estimates the value of a plant variable based on other plant measurements. Such sensor, estimating the C5 impurity at the top of a distillation tower is described in [1]. Refinary NOx emission modeled by an ANN is described in a recent paper [2].

An often-cited opposition to the use of ANN modeling in industrial diagnostics is the lack of "explanation" facility, the ability of the operator to understand the basis of the ANN recommendations. This paper shows that the "black-box" image of ANN model is misleading, and the trained ANN model can be analyzed to correctly explain the relationships between the plant operating variables and the NOx emission rate.

The structure of the paper is as follows: A brief description of the nitric acid plant, a review of the Guterman-Boger algorithms for large-scale ANN modeling, the Causal Index (CI) method of analyzing trained ANN, and the use of the hidden neurons' output values as clustering tool. Because of non-disclosure agreements, the exact details of the plant are withheld.

2 A Brief Description of the Nitric Acid Plant

The structure of the nitric acid plant is formed from two major units – a reactor in which ammonia gas is reacted with compressed air, resulting in the formation of nitrogen dioxide NO_2. The resulting high-temperature gaseous stream, that contain also the nitrogen, is used to make steam, and then is cooled before the second major unit, the absorption of the NO_2 by water in two absorption towers. As the reaction of the NO_2 with water results in some formation of nitrogen oxide, NO, that is not absorbed by water; additional air is fed to the first absorption tower to re-oxidize the NO to the absorbable NO_2. The equations governing the gas reactions and absorption are described in [3]. The gas exiting from the second absorption tower is sent to an expander, to utilize the high pressure, and then to the plant stack.

The allowed limit of the mixed nitrogen oxide species, NOx, in the "tail gas" was 400 ppm, and the plant was required to meet a reduced limit of 200 ppm. An additional reactor was needed to achieve this NOx concentration reduction, at great expense. Before deciding on this reactor, the plant management engaged the author to develop a model of the plant behavior, to find if changes in the plant operating variables would be able to achieve the required NOx emission reduction.

The chief plant operating engineer provided a database of 40 plant instrument measurements, saved every 5 minutes by the process computer, during the preceding six months. Included in this database was the measured NOx concentration in the tail gas that was to be the output of the ANN model.

3 A Short Overview of Artificial Neural Networks Modeling

An ANN model is trained by learning from known examples. A network of two layers of simple mathematical "neurons" is connected by weights. Data inputs are connected to the neurons in the first layer (called "hidden" neurons), which are connected in turn to the second layer of "output" neurons. Adjusting the values of the weights between the "neurons" during the training of the ANN is done by "back-propagation" of the errors between the output neurons and the known data outputs. Once the ANN is trained, it is verified by presenting examples not used in the training. The ANN may then be used to generate model outputs from the new examples presented to it. More information can be found in many books, such as [4] and journal articles, and in a review of the ANN literature, which is published in the comp.ai.neural-nets discussion group [5].

There are several obstacles in applying ANN to systems containing a large number of inputs and outputs. Most ANN training algorithms need thousands of repeated presentations ("epochs") of the training examples to finally achieve small modeling errors. Large ANN models tend to get stuck in local minima during the training. As most ANN training starts from random initial connection weight sets, and the number of neurons in the hidden layer are usually determined by heuristic rules, many re-training trials are needed to achieve good models. The Guterman-Boger (GB) training algorithm set [6] can easily train large scale ANN models, as it starts from non-random initial connection weights, obtained by the assumption that the inputs and outputs of the training data set are linearly related. The number of major PCA dimensions in the data recommends the number of hidden neurons (typically five), and the ANN is trained by the conjugate gradient method [7] with algorithms that avoid, and escape, local minima. It was found that even ANN with thousands of features could be trained in a matter of few hours on modern PC computers, even when the GB algorithm set is operating in the interactive MATLAB environment [8]. One of the algorithm set allows the identification of the more relevant features, and previous experience showed that a reduced dimensional ANN model is giving better results [9]. In this case, the number of features was small, 39, and the trained ANN gave good results, and thus no feature reduction was made.

Once a trained ANN is available, it can be analyzed for knowledge extraction. A causal index (CI) algorithm was proposed in [10] and found to be very useful in relating each input change influence on the relative magnitude and sign changes of each output [11]. The causal index method is an easily, somewhat qualitatively, method for rule extraction. The CI is calculated as the sum of the product of all "pathways" between each input to each output,

$$CI = \sum_{j=1}^{h} W_{kj} * W_{ji} \tag{1}$$

where there are $\boldsymbol{h}$ hidden neurons, $\mathbf{W_{kj}}$ are the connection weights from hidden neuron $\boldsymbol{j}$ to output $\boldsymbol{k}$, W_{ji} are the connection weights between input $\boldsymbol{i}$ to hidden neuron j.

Examining the CI for each output as a function of the inputs' number reveals the direction (positive or negative) and the relative magnitude of the relationship of the inputs on the particular output. Although somewhat heuristic, it is more reliable than local sensitivity checks. Their advantage is that they do not depend on a particular input vector, but on the connection weight set that represents all the training input vectors. This is also one of their limitations, as a local situation may be lost in the global representation.

Another useful analysis is to identify clusters in the data, is by training an unsupervised auto-associative ANN (AA-ANN), in which the features as presented both as inputs and outputs to the ANN. As there is no direct connections between the inputs and the outputs of the AA-ANN, and if the deviation between the real input feature vectors and the "predicted" input features is small, it means that the "binary" hidden neurons outputs are representing the essential information in the dataset in order to generate the correct outputs of the ANN model. It was found [11] that in a well-trained ANN, the hidden neurons' output values tend to be close to either one or zero. Thus they can be rounded into binary patterns, giving a maximum of 2h possible classes, if h is the number of hidden neurons. These "binary" values generate the minimum entropy (or the maximum information content) [12]. Thus, all data examples that generate the same hidden neurons output pattern are likely to belong to the same cluster. The values of features of each cluster are averaged and then divided by the average of the feature values of the full dataset. Feature ratios that are significantly different from unity are those that make each cluster distinct from other clusters. More information on the use of these techniques in industrial settings can be found in references [13,14].

4 ANN Model Training and Analyzing

The saved database, at 5 minutes interval collected in the January-July 2005 period, was cleaned by eliminating periods in which the plant operated at less than 100% capacity, or when alternative experimental operating policies were tried. In some cases when a process variable is measured by duplicate sensors, their readings were combined by averaging. The data were preprocessed by zero centering (subtracting the mean of each feature) and unit scaling (dividing by the standard deviation of each feature). The outputs of the ANN (and AA-ANN) models were further re-scaled into the [0.1 – 0.9] range. The two numbers of hidden neurons were selected, five and six, and the five hidden neurons model was found to give smaller modeling error.

Initially, the ANN model was trained with the 5 minute data to predict the NOx concentration. When it was found by the subsequent trained model analysis that the gas absorption temperature is one of the more important operating parameter affecting the NOx emission, an ANN model based on the daily averages was trained, thus eliminating the diurnal temperature change effect.

The 5 minute NOx concentration at the stack modeling results are shown in Figure 1. It can be seen that the mean average error between the actual measurements and the ANN model is 0.6%, with a standard deviation of 6.7%.

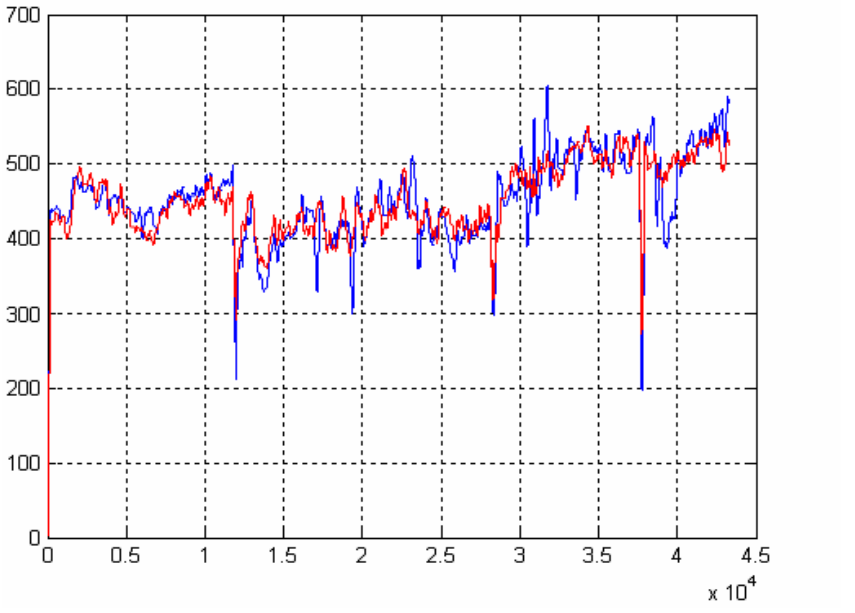


Fig. 1 ANN modeling of the 5 minute data. Mean relative error 0.6%, standard deviation 6.7%.Y scale - NOx concentration (ppm), Blue trace - measurements, Red trace ANN model output. X scale – sample number

The daily average ANN model was trained to give the total NOx emission, and the results are shown in Figure 2. The mean model error is 0.006 Kg/Hr NOx, with a standard deviation of 0.61 Kg/Hr.

The daily ANN model was analyzed by the Causal Index method. It was found that the NOx amount sent to the stack was positively dependant both on the reactor reactant flows, and the absorption tower temperatures. Both relationships are consistent with chemical engineering considerations. Some unexpected findings were found by the Causal Index values, but the reasons for these findings are explained in the Discussion section.

AA-ANN was then trained from the 5 minute data, presenting the preprocessed plant features (without the NOx measurements) both as inputs and outputs, again with five hidden neurons. After the training, the hidden neurons' outputs were rounded to one or zero, and the all data that had the same "binary" pattern were grouped into clusters.

When the full dataset was used to train the AA-ANN, 28 such clusters were identified, and the feature ratios results of the 22 clusters with non-trivial number of examples are shown in Table 1.

Some clusters (# 6, 8, 12, 13, 17, 27) have lower NOx emission amounts than the average NOx emission amount, and the identification of the feature ratios that are much smaller (or higher) then unity may explain these results. What was more

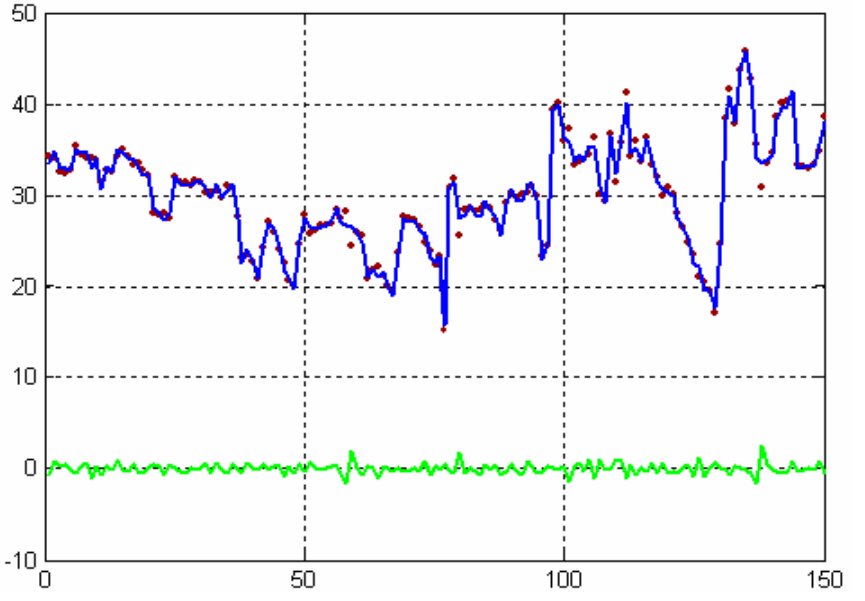


Fig. 2 ANN modeling of the daily average data of total NOx emission. X scale –day number, Y scale – Total NOx emission (Kg/Hr), Blue trace - measurements, Red dot - ANN model output, Green trace (bottom) - model-plant deviation (Kg/Hr)

Table 1 Feature ratios of the less polluting clusters

cluster #	6	8	12	13	17	27
# in cluster	1429	1374	1103	751	1627	105
KgNOx/hr	**21.23**	**23.71**	**17.89**	**21.18**	**20.04**	**20.16**
NFT1105	0.93	0.95	0.93	0.99	1.02	1.02
NFT1104	0.93	0.95	0.93	0.99	1.02	1.02
NFT1103	0.94	0.96	0.94	1.00	1.03	1.02
NFT1102	0.94	0.96	0.94	0.99	1.03	1.02
NTT1113	1.00	1.01	1.00	1.00	1.01	1.01
NTT1112	1.00	1.01	1.00	1.00	1.01	1.00
NT110050	0.99	1.00	0.99	1.00	0.99	0.98
NTT11007	0.99	1.00	0.99	1.00	0.99	0.99
NT110018	0.97	0.98	0.98	0.99	1.02	1.02
N2RATIO2	0.97	0.97	0.98	0.97	0.97	0.96
N2RATIO3	1.01	1.01	1.02	1.01	1.01	1.00
NTT11009	0.95	0.97	0.95	0.98	1.02	1.05
NFT1205	0.93	0.94	0.92	0.99	1.03	1.04

Table 1 (*continued*)

NTT1206	0.93	0.98	0.97	0.96	1.03	1.08
NT110010	0.93	0.98	0.96	0.98	1.02	1.05
NT110012	0.90	0.93	0.92	0.96	1.03	1.09
NT110011	0.90	0.93	0.92	0.96	1.03	1.09
NT110020	0.90	0.92	0.91	0.95	1.04	1.13
NFT1255	0.83	0.94	0.71	0.88	0.65	0.67
NPT1252	0.98	1.03	0.97	0.99	1.00	0.99
NTT1253	0.88	0.90	0.92	0.96	1.08	1.17
NPDT1250	1.00	0.87	0.94	1.09	0.96	0.97
NDPX1107	0.92	0.94	0.92	1.00	1.09	1.14
NAT1133	0.89	0.80	0.87	0.84	1.06	1.08
NAT1130	0.89	0.96	0.89	1.00	1.02	1.21
NTT1262	0.87	0.89	0.93	0.98	1.11	-0.52
NLT1258	1.04	1.04	1.03	1.00	1.01	1.03
NLT1208	1.03	1.03	1.03	1.03	1.06	1.21
NTT11004	0.87	0.89	0.94	0.98	1.11	1.08
NT110017	0.93	0.95	0.97	0.99	1.05	1.00
NTT11005	0.99	0.99	0.99	1.00	1.00	1.00
NTT701	0.99	0.99	0.99	1.00	1.00	1.03
NTT702	1.05	1.01	1.05	1.03	1.00	1.03
NT110046	1.03	0.97	1.03	1.04	1.01	1.00
NPT603	0.97	1.02	0.97	0.99	1.01	1.02
NFT1215A	0.93	0.95	0.93	0.99	1.02	1.02
NPT1101	0.98	0.96	0.95	1.03	1.01	0.98
NFT1206	0.95	0.94	0.95	1.01	1.00	0.68
N2NOX	**0.72**	**0.80**	**0.61**	**0.72**	**0.68**	**0.68**

surprising was the fact that some of the examples in these clusters are contiguous in time, very close to the duration of a complete shift (that is morning, afternoon or night shift). This raises the possibility that some shift managers are more efficient in running the plant and thus reducing the NOx emission.

5 Discussion

Reviewing the ANN modeling analysis revealed a major lack of information in the plant data collection scheme – no information on the control loop set points

values. These are changed by the shift managers, responding to plant upsets or transients. As these changes are not recorded, some cause and effects may be misunderstood. For instance, if the NOx absorption is seems insufficient, the set point of the absorbing water is increased. Subsequent analysis will relate high NOx emission with increased absorption water flow. Thus the daily feature averages is much reliable than 5 minute data.

If the control loop set points changes were available, the insight and experience of the better shift managers may be learned and incorporated in the computer control scheme, or at least known to the less experienced shift managers.

The major finding of the ANN modeling, that reducing the absorption tower operating temperature, was not helpful to solve the NOx emission issue, because the cooling water supply was outside the control of the plant management. Eventually, another NOx reducing technique was successfully adopted.

6 Conclusions

The ANN modeling of the nitric acid production plant predicted the NOx emission amounts and concentration in the tail gas with small errors. The analysis of the ANN and AA-ANN models revealed some known, and some previously unknown, relationships in the plant operation.

The ANN models trained from daily plant feature averages proved more informative than the 5 minute data, although it may be the results of the importance of the diurnal temperature changes in this plant.

The inclusion of the control loop set points in the plant database may provide more information for future analysis that will improve the operational knowledge for better efficiency.

Acknowledgements

Thanks are due to the plant chief operation engineer, Mr. A.R., for providing the plant data and helpful discussions, and to the plant management for allowing the publication of this paper.

References

1. Boger, Z., Guterman, H., Segal, T.: Application of large-scale artificial neural networks for modeling the response of a naphtha stabilizer distillation train. In: Proc. AIChE Ann Meeting, Chicago (1996)
2. Fortuna, L., Graziani, S., Xibilia, M.G.: Virtual instruments in refineries. IEEE Instr. Meas. Mag. 8(4), 26–34 (2005)
3. Sweeney, J.A., Liu, J.A.: Use of simulation to optimize NOx abatement by absorption and selective catalytic reduction. Ind. Eng. Chem. Res. 40(12), 2618–2627 (2001)
4. Bishop, C.M.: Neural Networks for Pattern Recognition. Clarendon Press, Oxford (1997)

5. Sarle, W.S.: Frequently asked questions. comp.ai.neural-nets users group (2005), ftp://ftp.sas.com/pub/neural
6. Guterman, H.: Application of principal component analysis to the design of neural networks. Neural Parallel Sci. Comput. 2, 43–54 (1994)
7. Leonard, J., Kramer, M.A.: Improvement of the back-propagation algorithm for training neural networks. Comput. Chem. Engng. 14, 337–341 (1990)
8. Boger, Z.: Who is afraid of the big bad ANN? In: Proc. Intl. Joint Conf. Neural Networks, pp. 2000–2005 (2002)
9. Boger, Z.: Selection of the quasi-optimal inputs in chemometric modeling by artificial neural network analysis. Anal. Chim. Acta 490(1-2), 31–40 (2003)
10. Baba, K., Enbutu, I., Yoda, M.: Explicit representation of knowledge acquired from plant historical data using neural network. In: Proc. Intl. Joint Conf. Neural Networks, vol. 3, pp. 155–160 (1990)
11. Boger, Z., Guterman, H.: Knowledge extraction from artificial neural networks models. In: Proc. IEEE Intl. Conf. Sys. Man Cyber., pp. 3030–3035 (1997)
12. Kamimura, R., Nakanishi, S.: Hidden information maximization for feature detection and rule discovery. Network Comput. Neural Sys. 6, 577–602 (1995)
13. Boger, Z.: Artificial neural networks modeling as a diagnostic and decision making tool. In: Ruan, D., Fantoni, P.F. (eds.) Power plant surveillance and diagnostics, modern approaches and advanced applications, vol. 16, pp. 243–252. Physica-Verlag, Heidelberg (2002)
14. http://optimalneural.com

Wavelet Neural Network as a Multivariate Processing Tool in Electronic Tongues

Juan Manuel Gutiérrez, Laura Moreno-Barón, Lorenzo Leija,
Roberto Muñoz, and Manel del Valle

Abstract. Electronic tongues are bioinspired systems that employ an array of sensors for analysis, recognition or identification of chemical species in liquids. This work presents the use of a Wavelet Neural Network (WNN) with multiple outputs to model a multianalyte quantification from a highly complex sensor signal. In this case, WNN accomplishes data reduction, feature extraction and modeling. WNN is implemented with a feedforward multilayer perceptron architecture, whose activation functions in its hidden layer neurons are wavelet functions, in our case, the first derivative of a Gaussian function. The neural network is trained using a backpropagation algorithm, adjusting the connection weights along with the wavelet parameters. The principle is applied to the simultaneous quantification of heavy metals present in a solution. Lead, Cadmium and Copper were therefore accurately determined in the 0.01-0.1ppm range in presence of Thallium and Indium with no need elimination of dissolved oxygen, as it is normally required in the standard chemical laboratory.

1 Introduction

Bioinspired technologies were developed to mimic the physiology of the mammal's senses in the design of analytical systems. Artificial vision, speech recognition, smell and taste, are just few examples in which the different areas of knowledge are working, in order to emulate the human abilities for linking the environmental characteristics to its corresponding conceptual sense. The basic foundation of these systems is the use of arrays of generic sensors, which offer low selectivity, and thus respond to most components contained in a sample. An electronic tongue is a chemical system that employs electrochemical sensors plus a chemometric processing tool, needed to decode the multivariate information present in the signal and interpret it in a consistent manner.

Juan Manuel Gutiérrez · Laura Moreno-Barón · Manel del Valle
Sensors & Biosensors Group, Dept. of Chemistry. Universitat Autònoma de Barcelona, Edifici Cn, 08193, Bellaterra, Spain

Lorenzo Leija · Roberto Muñoz
Bioelectronics Section, Department of Electrical Engineering, CINVESTAV, 07360 Mexico City, Mexico

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 73–81.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

In the case presented here, sensors are of voltammetric nature, i.e. those that generate a signal, a voltammogram, according to the existence of electrochemical reactions at a properly polarized electrode. In voltammetic electronic tongues, then, the nature of the signals involves the recording of currents generated for a scan of polarization potentials, using of a pair of electrodes. Voltammetric signals contain hundreds of readings and usually show overlapping regions with non-stationary characteristics [1], since all components in the solution are electrochemically active at a specific potential, and all them contribute to the current measure. Resolution and quantification of overlapping peaks in these records is a difficult task in electroanalysis [2].

Artificial Neural Network (ANN) is a powerful processing tool which has been used in electronic tongues. Processing voltammetric signals with ANN requires some kind of preprocessing stage for data reduction such as Principal Component Analysis (PCA), Discrete Fourier Transform (DFT) or Wavelet Transform (WT) in order to gain advantages in training time, to avoid redundancy in input data and to obtain a model with better generalization ability. This fact is mainly due to the extreme complexity and high dimension of these signals.

An innovative concept on ANN emerged in the last years known as Wavelet Neural Network (WNN) [3]. These networks have the main feature that the transfer function in neurons of the hidden layer is a mother wavelet. WNNs allow the feature extraction of the sensor signals while creating a multivariate calibration model, all in a single step, becoming an innovative chemometrics tool [1].

Cyclic Voltammetry (CV) has commonly been a qualitative technique in chemistry. However recent publications are introducing it coupled to chemometric techniques in order to quantify or semi-quantify the analytes of interest in a multivariate calibration approach [4-6].

This work develops a processing strategy that employs WNN to build multivariate models that describe the complexity of cyclic voltammograms, highly overlapped, in order to quantify properly five heavy metals (lead, copper, cadmium, thallium and indium) present in a solution.

2 Theory

The idea of combining wavelets with neural networks resulted in a successful synthesis of theories that generated a new class of networks called Wavelet Neural Network (WNN) [3]. These kind of networks use wavelet functions as hidden neuron activation functions. Using theoretical features of wavelet transform, network construction methods can be developed. The first approach to a WNN model can be inferred if the inversion formula for the Wavelet Transform (WT) is seen like the sum of the products between the wavelet coefficients and the family of daughter wavelets [7]. This definition established by Strömberg [8] replaces the corresponding integrals by a sum, therefore:

$$f(x) = \sum_{s=-\infty}^{\infty} \sum_{t=-\infty}^{\infty} w_{s,t} \psi_{s,t}(x) \tag{1}$$

where $w_{s,t}$ represent the wavelet coefficients of the decomposition of $f(x)$ and $\psi_{s,t}$ the daughter wavelets.

The WNN architecture has its fundamental principle on the similarity found between the inverse WT Strömberg's equation (1) and a hidden layer in the Multi-Layer Perceptron (MLP) network structure [8]; in the case of MLP, a WNN with only three layers (input, hidden and output layer) architecture is enough to approximate any arbitrary and continuous function, using an appropriate family of functions in the hidden layer [9-10]. The final accuracy in the approach depends of the used family function characteristics as well as by the error to reach.

For developing a WNN, wavelet frames are less complex to use than orthogonal wavelet basis. Frames can be constructed by simple operations of translation and dilation without fulfilling stringent orthogonal conditions [1,7,11]. Although many wavelet applications work well using orthogonal wavelet basis, many others work better with redundant wavelet families. The redundant representation offered by wavelet frames has demonstrated to be good both in signal denoising and in compaction [12-13].

In this manner, a signal $f(x)$ can be approximated by generalizing a linear combination of daughter wavelets $\psi_{s,t}(x)$ derived from its mother wavelet $\psi(x)$, this family of functions is defined as:

$$M_c = \left\{ \frac{1}{\sqrt{|s_i|}} \psi\left(\frac{x - t_i}{s_i} \right),\ t_i, s_i \in \Re,\ s_i > 0 \right\} \tag{2}$$

where the translation t_i and the scaling s_i are real numbers in $\Re$.

The family of functions M_c is known as a continuous wavelet frame of $L^2(\Re)$ if there exist two constants A and B that fulfills with [14]:

$$A\|f(x)\|^2 \le \sum_{s,t} \left| \langle f(x), \psi_i(x) \rangle \right|^2 \le B\|f(x)\|^2 \qquad \text{with} \quad A > 0,\ B < +\infty \tag{3}$$

Nevertheless for multi-variable model's applications it is necessary to use multi-dimensional wavelets. Families of multidimensional wavelets can be obtained from the product of P monodimensional wavelets, $\psi(a_{ij})$, of the form:

$$\Psi_i(x) = \prod_{j=1}^{P} \psi(a_{ij}) \qquad \text{where} \qquad a_{ij} = \frac{x - t_{ij}}{s_{ij}} \tag{4}$$

where t_i and s_i are the translation and scaling vectors respectively.

2.1 WNN Algorithm

The WNN architecture, shown in Fig. 1, corresponds to a feedforward MLP architecture with multiple outputs. The output $y^n(r)$ (where *n* is an index, not a power)

depends on the connection weights $c_i(r)$ between the output of each neuron and the r-th output of the network, the connection weights $w_j(r)$ between the input data and each output, an offset value $b_0(r)$ useful when adjusting functions that has a mean value other than zero, the n-th input vector x^n and the wavelet function Ψ_i of each neuron. The approximated signal of the model $y^n(r)$ can be represented by the next equation:

$$y^n(r) = \sum_{i=1}^{K} c_i(r)\Psi_i(x^n) + b_o(r) + \sum_{j=1}^{P} w_j(r)x_j^n \quad \text{with} \quad \{i,\ j,\ K,\ P\} \in Z \tag{5}$$

where $r = 1,2,...,m$, with $m \in Z$, represent the number of outputs, and subindexes i and j stand for the i-th neuron in the hidden layer and the j-th element in the input vector, x^n, respectively, K is the number of wavelet neurons and P is the length of input vector, x^n. With this model, a P-dimensional space can be mapped to a m-dimensional space ($R^P \rightarrow R^m$), allowing the network to predict a value for each output $y^n(m)$ when the n-th voltammogram x^n is input to the trained network.

The basic neuron will be a multidimensional wavelet $\Psi_i(x^n)$ which is built using the definition (4), where scaling (s_{ij}) and translation (t_{ij}) coefficients are the adjustable parameters of the i-th wavelet neuron. With this mathematical model for the wavelet neuron, the network's output becomes a linear combination of several multidimensional wavelets [3,15-17].

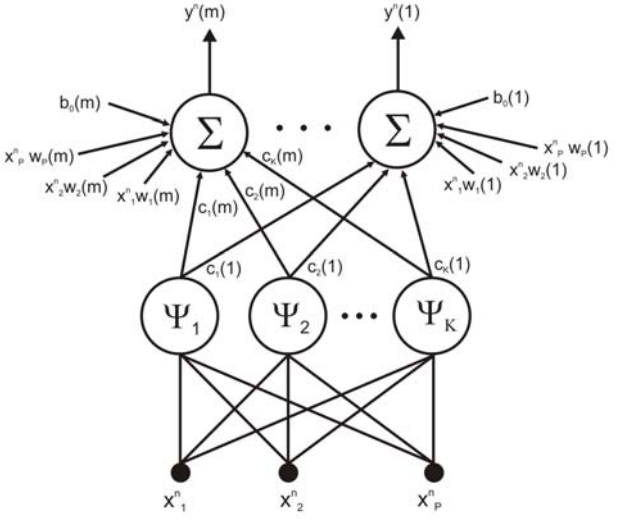


Fig. 1 Architecture of the implemented WNN

In the present work, the mother wavelet used as activation function corresponds to the first derivative of a Gaussian function defined by $\psi(x) = xe^{-0.5x^2}$. This function has demonstrated to be an effective choice for the implementation of WNN among others [18-20].

Once a network has been structured, a training strategy can be proposed. For this purpose we used error backpropagation; this method employs an iterative algorithm that looks for the minimum of the error function from the set of training vectors. In our application, the weights change once when all the vectors have been entered to the network (after one epoch). The difference is evaluated according to the *Mean Squared Error* (MSE) function defined by:

$$J(\Omega) = \frac{1}{2}\sum_{n=1}^{N}\sum_{r=1}^{m}\left(y_{\exp}^{n}(r) - y^{n}(r)\right)^{2} = \frac{1}{2}\sum_{n=1}^{N}\sum_{r=1}^{m}\left(e^{n}(r)\right)^{2} \tag{6}$$

where N is the number of input vectors, $y^n(r)$ is the *r*-th output of the network and $y_{\exp}^{n}(r)$ is the *r*-th real value related to the input vector x^n.

Because the proposed model is of multi-variable character, we define:

$$\Omega = \left\{b_0(r), w_j(r), c_i(r), t_{ij}, s_{ij}\right\} \tag{7}$$

as the set of parameters that will be adjusted during training.

These parameters must change in the direction determined by the negative of the output error function's gradient:

$$-\frac{\partial J}{\partial \Omega} = \frac{1}{N}\sum_{n=1}^{N}\sum_{r=1}^{m} e^{n}(r)\frac{\partial y^{n}(r)}{\partial \Omega} \tag{8}$$

In order to reduce training time, the value 1/N in (8) is a term that averages the error with the number of inputs vectors. The index *m* corresponds to the number of outputs.

The changes in network parameters are calculated at each iteration according to $\Delta\Omega = -\mu\frac{\partial J}{\partial \Omega}$, where μ is a positive real value known as *learning rate*. With these changes the variables contained in Ω are updated using:

$$\Omega_{new} = \Omega_{old} + \Delta\Omega \tag{9}$$

where Ω_{old} represents the current values and $\Delta\Omega$ represents the changes.

On the one hand, initialization of network parameters was done according to a previous reported methodology [1,21], where the weights are initialized with random initial values, and the parameters of scale and translation are proposed in order to avoid the concentration of wavelets in localities of the input data universe.

On the other hand the algorithm has two conditions to stop the training process when any of them is accomplished. These conditions are a maximum limit of training epochs and the convergence error threshold.

3 Application in Chemical Sensing

The principles of the electronic tongue are applied in order to solve a mixture of five heavy metals by direct cyclic voltammetric analysis. The approach uses the voltammetric signal for the resolution of the components involved, a platinum modified graphite-epoxy composite, acting as the working electrode; and a multivariate calibration model built using WNN.

This study corresponds to the direct multivariate determination of lead, copper, cadmium, thallium and indium from the complete cyclic voltammograms of the complex matrix. A total of 30 pattern solutions were prepared in acetate buffer (0.1 M, pH 3.76) for the generation of the response model. The five metals are present in all patterns in different concentrations which vary from 0.01 to 0.1ppm.

A commercial potentiostat (Autolab PGSTAT 30) with the home made working electrode was used for the measurements. The cell was completed with a Ag/AgCl reference electrode (Orion 900200) and a commercial Platinum counter electrode (model 52-67 1, Crison). All experiments were carried out without any oxygen removal from the sample, which is a common great interference in electroanalysis. ºCyclic voltammetry signal is obtained by applying a linear potential sweep to the working electrode; once it reaches a set potential, the sweep direction is reverted. In this way, each resulting voltammogram consisted of 534 current values recorded in a range of potentials from -1.0V to 0.3V to -1.0V in steps of 0.0048V. For the prediction model a five-output WNN with three neurons in the hidden layer was programmed and trained. The input layer used 534 neurons, defined as the width of the voltammograms (oxidation and reduction sweeps in the CV signal). The expected output error was programmed to reach a value of 0.025 ppm, evaluated by $2/N \cdot J(\Omega)$, where $J(\Omega)$ is the MSE defined in (6); and the learning rate was set to 0.0004.

4 Results and Discussion

In order to resolve individual analytes from the measured CV voltammograms, WNN models were programmed for the simultaneous quantification of the five metal. WNN modeling performance was evaluated according with a Linear Regression Analysis of the comparations between the expected concentration values and those obtained for training and testing sets. All of them evaluated for a 95 % interval of confidence. Table 1, shows training and testing process for one programmed model. All quantification process were successfully accomplished , showing the good characteristics of the WNN for modeling non-linear input-output relationships. The slope (m) and intercept (b) which defines the comparison line y=mx+b that best fits the data (along with the uncertainly interval for a 95% of significance) are shown for one of the study cases The ideal case implies lines with m=1 and b=0, which is fulfilled in all cases at the appropriate confidence level.

Table 1 Comparison lines plotted to the results of the training and testing data using a WNN architecture

Metal	Training					Testing				
	R	m	error	b	Error	R	m	error	b	error
Pb	0.986	0.994	±0.084	-3.979E-05	±0.006	0.899	1.188	±0.472	-4.795E-03	±0.034
Cd	0.964	0.997	±0.135	-7.704E-04	±0.010	0.923	1.361	±0.462	-1.365E-02	±0.034
Cu	0.960	0.979	±0.141	9.791E-04	±0.010	0.892	1.240	±0.514	-5.192E-03	±0.037
Tl	0.983	0.992	±0.093	3.434E-04	±0.007	0.745	0.654	±0.477	1.445E-02	±0.035
In	0.949	1.011	±0.166	-1.357E-03	±0.011	0.837	1.028	±0.548	-2.419E-03	±0.037

Five extra WNNs were trained to confirm the model's performance. Using a random selection each time (following a 10-fold cross validation process), to form different sets of training and testing. The behavior of different models was evaluated averaging of the Recovery Percentage (RP) between the expected and obtained concentration values .RP value is a parameter which indicates the ability of the model to determine quantitatively a chemical analyte present in a sample; the ideal value is 100% [1]. Figure 2 summarizes the results for all metal species present in the sample which could be quantified by WNN models.

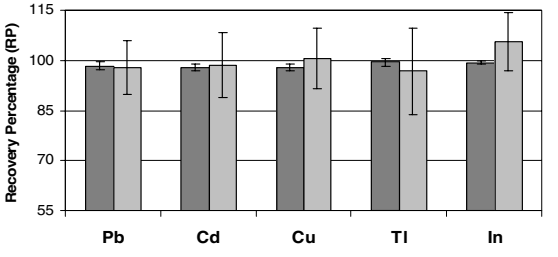


Fig. 2 Recovery Percentage average for Training (dark grey) and Testing (light grey) subsets. The lines on the bars, correspond to the standard deviation of the five replicates

5 Conclusions

Based on the results, it is possible to observe that the simultaneous quantitative determination of metallic species was achieved successfully employing a WNN model. From the estimated concentration values, it is possible to conclude that the estimation of metals of interest (Pb, Cd and Cu) at the sub-ppm level and in the presence of interfering Tl and In, was attained, with average errors lower than 5%.

The proposed approach has demonstrated to be a proper multivariate modelling tool for voltammetric analytical signals. For its operation, the WNN adjusts the parameters for a family of wavelet functions that best fits the shapes and frequencies of sensors' signals. Moreover, cyclic voltammetry, commonly being a qualitative technique, proved to be a tool that brought satisfactory results in elec-

trochemical quantification of heavy metals. WNNs were able to extract meaningful information from the signal in order to estimate properly different species concentrations, even in the presence of important interfering elements as oxygen.

Should be noted as well, that there was no need of surface renewal of the platinum composite working electrode to obtain proper results; all these features confer clear advantages for the development of autonomous chemical surveillance systems.

Acknowledgements

Financial support for this work was provided by CONACYT (México) through the project "Apoyos Vinculados al Fortalecimiento de la Calidad del Posgrado Nacional, a la Consolidación de Grupos de Investigación, y de la Capacidad Tecnológica de las Empresas. Vertiente II". Mobility of researchers was possible through Spanish Agency AECI within the framework of the project A/012192/07 (AMALENG).

References

[1] Gutés, A., Céspedes, F., Cartas, R., Alegret, S., del Valle, M., Gutiérrez, J.M., Muñoz, R.: Multivariate calibration model from overlapping voltammetric signals employing wavelet neural network. Chemometr. Intell. Lab. Syst. 83, 169–179 (2006)

[2] Nie, L., Wu, S., Wang, J., Zheng, L., Lin, X., Rui, L.: Continuous wavelet transform and its application to resolving and quantifying the overlapped voltammetric peaks. Anal. Chim. Acta. 450, 185–192 (2001)

[3] Zhang, Q., Benveniste, A.: Wavelet Networks. IEEE Trans. Neural. Netw. 3, 889–898 (1992)

[4] Saurina, J., Hernández-Cassou, S., Fàbregas, E., Alegret, S.: Cyclic voltammetric simultaneous determination of oxidizable amino acids using multivariate calibration methods. Anal. Chim. Acta. 405, 153–160 (2000)

[5] Kramer, K.E., Rose-Pehrsson, S.L., Hammond, M.H., Tillett, D., Streckert, H.H.: Detection and classification of gaseous sulfur compounds by solid electrolyte cyclic voltammetry of cermet sensor array. Anal. Chim. Acta. 584, 78–88 (2007)

[6] Moreno-Baron, L., Cartas, R., Merkoçi, A., Alegret, S., Gutiérrez, J.M., Leija, L., Hernández, P.R., Muñoz, R., del Valle, M.: Data Compression for a Voltammetric Electronic Tongue Modelled with Artificial Neural Networks. Anal. Lett. 38, 2189–2206 (2005)

[7] Akay, M.: Time Frequency and wavelets. In: Akay, M. (ed.) Biomedical Signal Processing. IEEE Press Series in Biomedical Engineering. Wiley-IEEE Press (1992)

[8] Meyer, Y.: Wavelets, Algorithms and Applications. Society for Industrial and Applied Mathematics, 2nd edn. SIAM, Philadelphia (1994)

[9] Hornik, K.: Multilayer feedforward networks are universal approximators. Neural. Netw. 2, 359–366 (1989)

[10] Scarcelli, F., Tsoi, A.C.: Universal Approximation using Feedforward Neural networks: A survey of some existing methods and some new results. Neural. Netw. 11, 15–37 (1998)

[11] Heil, C.E., Walnut, D.F.: Continuous and discrete wavelet transforms. SIAM Review 31, 628–666 (1989)

[12] Daubechies, I., Grossmann, A., Meyer, Y.: Painless nonorthogonal expansions. J. Math. Phys. 27, 1271–1283 (1986)
[13] Daubechies, I.: Ten Lectures on wavelets. CBMS-NSF Regional Conference Series In Applied Mathematics, vol. 61. Society for Industrial and Applied Mathematics, Philadelphia (1992)
[14] Kugarajah, T., Zhang, Q.: Multidimensional wavelet frames. IEEE Trans. Neural Netw. 6, 1552–1556 (1995)
[15] Cannon, M., Slotine, J.E.: Space-frequency localized basis function networks for nonlinear system estimation and control. Neurocomputing 9, 293–342 (1995)
[16] Mallat, S.: A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans. Pattern. Anal. Mach. Intell. 11, 674–693 (1989)
[17] Zhang, J., Walter, G.G., Miao, Y., Lee, W.N.W.: Wavelet neural networks for function learning. IEEE Trans. Signal Process. 43, 1485–1497 (1995)
[18] Guo, Q.X., Liu, L., Cai, W.S., Jiang, Y., Liu, Y.C.: Driving force prediction for inclusion complexation of α-cyclodextrin with benzene derivates by a wavelet neural network. Chem. Phys. Lett. 290, 514–518 (1998)
[19] Zhang, X., Qi, J., Zhang, R., Liu, M., Hu, Z., Xue, H., Fan, B.: Prediction of programmed-temperature retention values of naphthas by wavelet neural networks. Comput. Chem. 25, 125–133 (2001)
[20] Zhong, H., Zhang, J., Gao, M., Zheng, J., Li, G., Chen, L.: The discrete wavelet neural network and its application in oscillographic chronopotentiometric determination. Chemometr. Intell. Lab. Syst. 59, 67–74 (2001)
[21] Oussar, Y., Rivals, I., Personnaz, L., Dreyfus, G.: Training wavelet networks for nonlinear dynamic input-output modeling. Neurocomputing 20, 173–188 (1998)

Design of ANFIS Networks Using Hybrid Genetic and SVD Method for the Prediction of Coastal Wave Impacts

Ahmad Bagheri, Nader Nariman-Zadeh, Ali Jamali, and Kiarash Dayjoori

Abstract. Genetic Algorithm (GA) and Singular Value Decomposition (SVD) are deployed respectively for optimal design of Gaussian membership functions of antecedents, and linear coefficients vector of consequents, in ANFIS (Adaptive Neuro-Fuzzy Inference Systems) networks used for prediction of the 3rd month ahead coastal wave impacts. For this purpose time series parameters chosen by GMDH (Group Method of Data Handling) modeling are utilized. The aim of such modeling is to demonstrate how ANFIS network can be coupled with GMDH network and to show how such combination by hybrid GA/SVD designed ANFIS networks result in precise models.

1 Introduction

System identification techniques are applied in many fields in order to model and predict the behaviors of unknown and/or very complex systems based on given input-output data. Theoretically, in order to model a system, it is required to understand the explicit mathematical input-output relationship precisely. Such explicit mathematical modeling is, however, very difficult and is not readily tractable in poorly understood systems. Alternatively, soft-computing methods, which concern computation in imprecise environment, have gained significant attention. Fuzzy rule-based systems have been an active research field of the soft-computing methods because of their unique ability. Indeed, a fuzzy-logic system is able to model approximate and imprecise reasoning processes common in human thinking or human problem solving.

Among fuzzy models, the Takagi-Sugeno-Kang type fuzzy models, also known as TSK models, are widely used for control and modeling because of their high accuracy and relatively small models. In the TSK models, which are also known as Neuro-Fuzzy systems, the consequents of the fuzzy rules are explicit functions, usually linear function, of the input variables rather than fuzzy sets [1]. An

Ahmad Bagheri · Nader Nariman-Zadeh · Ali Jamali · Kiarash Dayjoori
Department of Mechanical Engineering, Faculty of Engineering,
The University of Guilan, P.O. Box 3756, Rasht, Iran
e-mail: bagheri@guilan.ac.ir

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 83–92.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

equivalent approach to the TSK models has been proposed as Adaptive Neuro-Fuzzy Inference System (ANFIS). In ANFIS networks a hybrid learning method is used for tuning parameters in both antecedents and consequents parts of embodied TSK-type fuzzy rules. There have been some research efforts in the literature to optimally design the premise and conclusion parts of such ANFIS or TSK models. In some very recent works genetic algorithm is used in conjunction with SVD for determination of nonlinear and linear parameters embodied in the antecedent and consequent parts of fuzzy rules [2].

Unique and desirable properties of fuzzy logic and ANFIS modeling have caused their wide application in climatic and meteorological studies. Among climatic natural consequences, Wind produced waves are of exceptional interest to engineers due to their impact on coastal structures. Since waves impact determination comprises of assigning uncertain or complex interaction between elements and parameters, and since long-term observed data are inadequate in characteristic of wind produced waves there have been considerable studies devoted to their main properties prediction by soft computing methods [3].

In this paper, third month ahead wave impacts of a town located on south-east coast of Caspian Sea which is called NosratAbad, is predicted by optimum ANFIS networks. Such ANFIS networks are optimized by simultaneous application of genetic algorithm and SVD method. Genetic algorithm determines optimal Gaussian membership functions of rules premise part, whilst SVD method selects linear parameters of the rules' conclusion part. In order to distinct parameters of time series which were involved for wave impacts forecasting, an identical prediction was fulfilled by GMDH (Group Method of Data Handling) network and selected parameters were filled as input to ANFIS network. For reducing the complexity of the rule base, a "bottom up" rule-based approach [4] is adopted to search for structures with the best number of rules and prediction errors.

2 Study Area

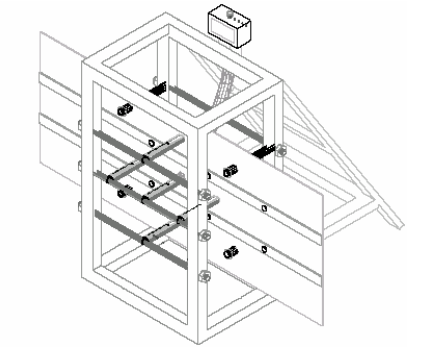
The data set used in this study is average of maximum wave impacts per month gathered in NosratAbad town on south-west coast of Caspian Sea at 38º 24′ N and 40º E, from January 1996 to December 2000. Measuring apparatus, which will be discussed later, was located 10m from the shore line and above the sea natural surface where depth of the sea was 30cm. Wave impacts were collected for 45 minutes at 9, 15 and 21 local time of each day [5].

Since the Caspian Sea is located in middle-latitude and is separated from world oceans, its properties are mostly influenced by seasonal changes [6, 7]. Thus it could be expected that its natural phenomena, particularly water wave impacts, usually occur in shorter period compared to others worldwide natural phenomena and interprets utilizing of Time Series with short term data for assigning water wave attributes.

3 Basis of Design and Measuring Manner of Sea Wave Impact Tester

As it is shown in figure (1), the apparatus consists of a six sides frame with an iron plate of $0.3\text{cm}\times40\text{cm}\times100\text{cm}$ in the centre. The plane is reinforced by two iron band so that buckling at the time of waves' impact is prevented. There are five holes on the plane for installing and bolting of force measures. One of these holes is located at the centre of the plane and the remaining four are made at corners of an abstract square which its centre coincides with the plan centre. Five force measurers with capability of 12.5 kgf are mounted perpendicular to the plane of the instrument. For balancing the wave breaker reciprocation four circular profiles are installed on both back sides of frame. Each measurer's spring stiffness coefficient is equal to 2041.66 N/m Therefore, by consideration of springs' parallelism the total springs' stiffness is equal to 10208.3 N/m. Force counter sensor, which is paralleled with force measurers, records the received force in kgf unit with precision of one grf.

Fig. 1 Wave impact measurer apparatus



By positioning wave impact measurer on seashore, the plane reacts as a wave breaker and it is pushed during the wave impacts. When the plane is pushed, force-measurers are stretched and their elongation indicates applied force [5].

4 Modelling Using ANFIS

An ANFIS that consists of a set of TSK-type fuzzy IF-THEN rules can be used in modelling in order to map inputs to outputs. The formal definition of such identification problem is to find a function $\hat{f}$ so that it can be approximately used instead of the actual one, f, in order to predict output $\hat{y}$ for a given input vector $X=(x_1,x_2,x_3,...,x_n)$ as close as possible to its actual output y or

$$\sum_{i=1}^{m}[\hat{f}(x_{i1},x_{i2},x_{i3},...,x_{in})-y_i]^2\rightarrow\min. \tag{1}$$

In this way, a set of linguistic TSK-type fuzzy IF-THEN rules is designed to approximate f by $\hat{f}$ using m observations of n-input–single-output data pairs (X_i, y_i) (i=1, 2, …, m). The fuzzy rules embodied in such ANFIS models can be conveniently expressed using the following generic form

$$\text{Rule}_l : \text{IF} \quad x_1 \text{ is } A_l^{(j_1)} \text{ AND } x_2 \text{ is } A_l^{(j_2)} \text{ AND},$$
$$..., x_n \text{ is } A_l^{(j_n)} \text{ THEN } y = \sum_{i=1}^{n} w_i^l x_i + w_0^l \tag{2}$$

in which $j_i \in \{1, 2, ..., r\}$, and $W^l = \{w_1^l, w_2^l, ..., w_n^l, w_0^l\}$ is the parameter set of the consequent of each rule. The entire fuzzy sets in x_i space are given as

$$A^{(i)} = \{A^{(1)}, A^{(2)}, A^{(3)}, ..., A^{(r)}\}. \tag{3}$$

These entire fuzzy sets are assumed Gaussian shape defined on the domains $[-\alpha_i, +\beta_i]$ (i=1,2,…,n). The fuzzy sets are represented by Gaussian membership functions in the form of

$$\mu_{A^{(j)}}(x_i) = Gaussian\,(x_i; c_j, \sigma_j) = \exp\left(-\frac{1}{2}\left(\frac{x_i - c_j}{\sigma_j}\right)^2\right) \tag{4}$$

where c_j, σ_j are adjustable centers and variances in antecedents, respectively. It is evident that that the number of such parameters involved in the antecedents of ANFIS models can be readily calculated as nr, where n is the dimension of input vector and r is the number of fuzzy sets in each antecedent. Using Mamdani algebraic product implication, the degree of such local fuzzy IF-THEN rule can be evaluated in the form

$$\mu_{Rule_l} = \prod_{i=1}^{n} \mu_{A_l^{(i)}}(x_i). \tag{5}$$

In this equations $\mu_{A_l^{(ji)}}(x_i)$ represents the degree of membership of input x_i regarding their lth fuzzy rule's linguistic value, A_l^{ji}. Using singleton fuzzifier, product inference engine, and finally aggregating the individual contributions of rules leads to the fuzzy system in the form

$$f(X) = \frac{\sum_{l=1}^{N} y_l \left(\prod_{i=1}^{n} \mu_{A_l^{(ji)}}(x_i)\right)}{\sum_{l=1}^{N} \left(\prod_{i=1}^{n} \mu_{A_l^{(ji)}}(x_i)\right)}. \tag{6}$$

when a certain set containing N fuzzy rules in the form of equation (2) is available. Equation (6) can be alternatively represented in the following linear regression form

$$f(X) = \sum_{l=1}^{N} p_l(X) y_l + D \tag{7}$$

where D is the difference between $f(X)$ and corresponding actual output, y, and

$$p_l(X) = \frac{(\prod_{i=1}^{n} \mu_{A_l^{(ji)}}(x_i))}{\sum_{l=1}^{N} (\prod_{i=1}^{n} \mu_{A_l^{(ji)}}(x_i))} \tag{8}$$

It is therefore evident that equation (7) can be readily expressed in a matrix form for a given m input-output data pairs (X_i, y_i) (i=1, 2, …, m) in the form

$$Y = P.W + D, \tag{9}$$

where $W = [w_1, w_2, ..., w_s]^T \in \Re^S$, S=N(n+1) and $P = [p_1, p_2, ..., p_s]^T \in \Re^{m \times S}$. It should be noted that each (n+1) components of vector w_i corresponds to the conclusion part of a TSK-type fuzzy rule. Such firing strength matrix P is obtained when input spaces are partitioned into certain number of fuzzy sets. It is evident that the number of available training data pairs is usually larger than all the coefficients in the conclusion part of all TSK rules when the number of such rules is sufficiently small, that is $m \geq S$. This situation turns the equation (9) into a least squares estimation process in terms of unknowns, $W = [w_1, w_2, ..., w_s]^T$, so that the difference$D$ is minimized. The governing normal equations can be expressed in the form

$$W = (P^T.P)^{-1}.P^T.Y \tag{10}$$

Such modification of coefficients in the conclusion part of TSK rules leads to better approximation of given data pairs in terms of minimization of difference vector D. However, such direct solution of normal equations is rather susceptible to round off error and, more importantly, to the singularity of these equations [2].

Therefore singular value decomposition as a powerful numerical technique could be used to optimally determine the linear coefficients embodied in the conclusion part of ANFIS model to deal with probable singularities in equation (9). However, in this work, a hybridization of genetic algorithm and SVD is proposed to optimally design an ANFIS network for prediction of the NosratAbad coastal wave impacts. Such combination of genetic algorithms and SVD is described in sections 5 and 6, respectively.

5 Application of Genetic Algorithm to the Design of ANFIS

The incorporation of genetic algorithm into the design of such ANFIS models starts by representing the nr real-value parameters of $\{c_j, \sigma_j\}$ as a string of concatenated sub-strings of binary digits and selected rules as a string of decimal digits in interval of $\{1, nr\}$. Thus, combination of the binary and the decimal strings

represents antecedent part of a fuzzy system. Fitness, Φ, of Such ANFIS system to model and predict wave impacts, is readily evaluated in the form of

$$\Phi = 1/E \tag{11}$$

where E is the objective function given by equation (1) and is minimized through an evolutionary process by maximization the fitness, Φ. The evolutionary process starts by randomly generating an initial population of binary and decimal strings each as a candidate solution representing the fuzzy partitioning and the rules of the premise part. Then, using the standard genetic operations of roulette wheel selection, crossover, and mutation, entire populations of combined strings are caused to improve gradually. In this way linear coefficients of conclusion parts of TSK rules corresponding to each chromosome representing the premise parts of the fuzzy system, are optimally determined by using SVD. Therefore, ANFIS models of predicting coastal wave impacts with progressively increasing fitness, Φ, are produced whilst their premise and conclusion parts are determined by genetic algorithms and SVD, respectively. In the following section, a brief detail of SVD application for optimally determination of consequent coefficients is described [2].

6 Application of Singular Value Decomposition to the Design of ANFIS Networks

In addition to the genetic learning of antecedents of fuzzy systems involved in ANFIS networks, singular value decomposition is also deployed for the optimal design of consequents of such fuzzy systems. Singular value decomposition is the method for solving most linear least squares problems that some singularities may exist in the normal equations. The SVD of a matrix, $P \in \Re^{M \times S}$, is a factorisation of the matrix into the product of three matrices, a column-orthogonal matrix $U \in \Re^{M \times S}$, a diagonal matrix $Q \in \Re^{S \times S}$ with non-negative elements (singular values), and an orthogonal matrix $V \in \Re^{S \times S}$ such that

$$P = U.Q.V^T \tag{12}$$

The most popular technique for computing the SVD was originally proposed in [8]. The problem of optimal selection of W in equation (9) is firstly reduced to finding the modified inversion of diagonal matrix Q in which the reciprocals of zero or near zero singulars (according to a threshold) are set to zero. Then, such optimal W is obtained using the following relation [2]

$$W = V\,[\mathrm{diag}(1/q_j)]\,U^T\,Y \tag{13}$$

7 Genetic/SVD Based ANFIS Prediction of Coastal Wave Impacts

The 60 experimental data which are used in this study are monthly average of daily maximum wave impacts from January 1996 to December 2000. However, in order

to construct an input-output table to be used by ANFIS network, a pre-table of 60 various input have been considered for possible contribution to represent the model. The first 15 columns of such pre-input-output data table consist of the wave impacts in the 1st, …, 15th previous months denoted by *impact*(*i-1*) , …, *impact*(*i-15*) , respectively. The next 15 columns of it consist of the increment values, denoted by *Inc_1*(*i*), …, *Inc_j*(*i*),…, *Inc_15*(*i*),which is defined as

$$Inc_j(i) = impact(i-j) - impact(i-j-1) \tag{14}$$

where *i* is the index of current month and *j* is the index of a particular increment. The next 15 columns consist of moving averages of previous month impact denoted by *MA_I_2*(*i*), …, *MA_I_j*(*i*),…, *MA_I_16*(*i*), which is defined as

$$MA_I_j(i) = \sum_{k=1}^{j} \frac{impact(i-k)}{j} \tag{15}$$

where *i* is the index of current month and *j* is the index of a particular moving average of wave impacts. The last 15 columns of the pre-input-output data table consist of moving averages of previous month increments denoted by *MA_Inc_2*(*i*), …, *MA_Inc_j*(*i*),…, *MA_Inc_16*(*i*) which is defined as

$$MA_Inc_j(i) = \sum_{k=1}^{j} \frac{Inc_k(i)}{j} \tag{16}$$

where *i* is the index of current month and *j* is for the index of a particular moving average of increments. In the final stage of pre-data-table creation, wave impacts of the 3rd month ahead were substituted as outputs. Since, such aforementioned method is used for creation of pre-input-output-data-table, 41 observation points are available for the prediction of the 3rd month ahead wave impact [7].

In order to predict wave impact of the 3rd month ahead, pre-data-table was randomly divided into 30-member training and 11-member testing sets. Because of ANFIS networks deficiency in input data selection, same prediction was deployed by GMDH networks in order to distinct main inputs to ANFIS network. Chosen input by GMDH network which were 15th increment or Inc_15(i), the 10th moving average of wave impacts or MA_I_11(i), wave impact of the 5th pervious month or impact(i-5) and the 7th moving average of increments or MA_Inc_8(i), were filled to discussed GA/SVD designed ANFIS network for prediction of the 3rd month ahead wave impact. The prediction was deployed with 2, 3 and 4 membership functions in space of each input and the number of rules was gradually extended from 2 to 16 numbers in every case in order to determine the optimum number of rules. In developing of such ANFIS networks, training mean square error (MSE) was considered as the objective function of the consequent or SVD designed part of the ANFIS system and the testing MSE was regarded as the objective function of its premise or GA designed part. Training and testing MSEs of the ANFIS networks with two membership functions in space of each input and different numbers of rules, are plotted in figure (2).

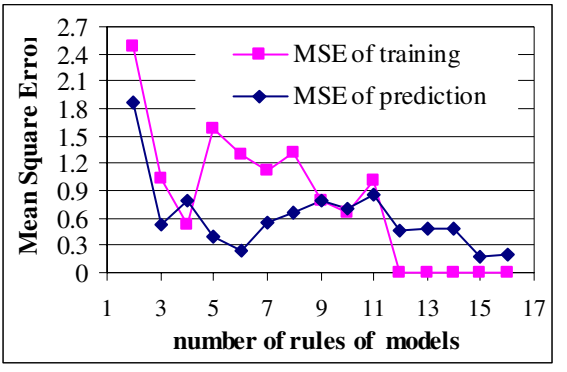


Fig. 2 The training and prediction MSEs of wave impact predictions with ANFIS networks of two membership functions in space of each input

From figure 2, it is perceived that ANFIS network with 4 numbers of rules is the best choice in terms of training error, testing error and number of rules. Thus the ANFIS network of 4 rules with training MSE of 0.57 and testing MSE of 0.75 was chosen as the final model for the prediction of 3rd month ahead wave impacts. Membership functions of such ANFIS network are represented in table 1, where c_j, σ_j are adjustable centers and variances given by equation (4).

Table 1 Genetically evolved Gaussian membership functions of the ANFIS network developed for prediction of the 3rd month ahead wave impacts

Inc_15(i)	*c*	*σ*
A_1	-4.4	1.69
A_2	-5.2	3.67

MA_I_11(i)	*c*	
B_1	6.94	4.73
B_2	6.3	1.95

impact(i-5)	*c*	*σ*
C_1	2.03	5.58
C_2	2.03	9.68

MA_Inc_8(i)	*c*	*σ*
D_1	-0.34	0.77
D_2	-0.34	0.51

The set of TSK-type fuzzy rules obtained for the prediction of the 3rd month ahead wave impacts are presented in equations (17) to (20), where x_1, x_2, x_3 and x_4 stands for *Inc_15(i)*, *MA_I_11(i)*, *impact(i-5)* and *MA_Inc_8(i)* respectively. It should be noted that the number of parameters in each vector of coefficients in the conclusion part of every TSK-type fuzzy rule is equal to 5 according to the assumed linear relationship of input variables in the consequents.

$$\text{if } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_2 \text{ and } x_3 \text{ is } C_2 \text{ and } x_4 \text{ is } D_2, \text{then } impact(i+2) = -8.2655\times x_1 + 50.069\times x_2 - 0.037476\times x_3 + 13.084\times x_4 - 306.1 \quad (17)$$

$$\text{if } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_1 \text{ and } x_3 \text{ is } C_1 \text{ and } x_4 \text{ is } D_1, \text{then } impact(i+2) = -121.83\times x_1 - 302.81\times x_2 - 21.316\times x_3 + 110.61\times x_4 + 2177.1 \quad (18)$$

$$\text{if } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ and } x_3 \text{ is } C_1 \text{ and } x_4 \text{ is } D_1, \text{then } impact(i+2) = -115.41\times x_1 - 301.19\times x_2 - 16.348\times x_3 + 114.95\times x_4 - 2148 \quad (19)$$

$$\text{if } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ and } x_3 \text{ is } C_2 \text{ and } x_4 \text{ is } D_1, \text{then } impact(i+2) = -6.7227\times x_1 - 0.98431\times x_2 + 0.90803\times x_3 - 0.43452\times x_4 + 19.88 \quad (20)$$

The very good behavior of the ANFIS network designed by hybrid GA/SVD to model and predict the wave impact of the 3rd month ahead is depicted in Figure (3).

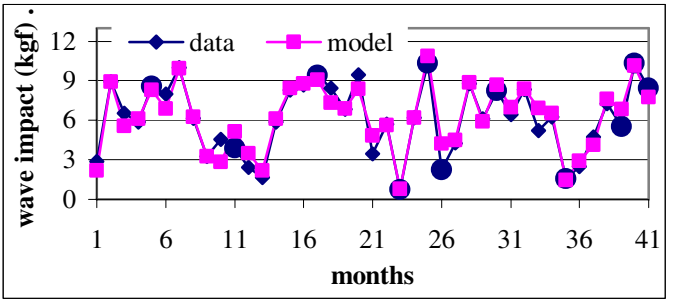


Fig. 3 Comparison of the actual data and the behavior of the ANFIS network (data points distinct by circles represent testing set members; remaining points comprise training set)

8 Conclusion

It has been shown that Hybrid GA/SVD-designed ANFIS networks provide effective means to model and predict the 3rd month ahead wave impacts in terms of training and testing errors and the number of rules, with training error of 8.36%, testing error of 10.35% and rule numbers of four which are quite acceptable errors and number of rules for meteorological phenomena forecasting. Additionally it was demonstrated that hybridization of SVD with GA despite other liner equation solving methods ensure model convergence for any given antecedents part of an ANFIS network. These have been achieved by utilization of time series and inputs selected by GMDH networks. It has been also shown that usage of GMDH network results as input to ANFIS network contribute to TSK-type fuzzy system the ability of time series modelling and result in accurate models.

References

1. Hoffmann, F., Nelles, O., et al.: Genetic programming for model selection of TSK-fuzzy systems. J. Information Sciences 136, 7–28 (2001)
2. Nariman-zadeh, N., Darvizeh, A., Dadfarmai, M.H., et al.: Design of ANFIS networks using hybrid genetic and SVD methods for the modelling of explosive cutting process. J. Materials Processing Technology 155-156, 1415–1421 (2004)
3. Kazeminezhad, M.H., Etemad-Shahidi, A., Mousavi, S.J., et al.: Application of fuzzy inference system in the prediction of wave parameters. J. Ocean Eng. 32, 1709–1725 (2005)
4. Mannle, M., et al.: Identifying Rule-Based TSK fuzzy Models. In: Proceeding of 7th European Congress on Intelligent Techniques and Soft Computing (1999)

5. Bagheri, A., Tavoli, M.A., Karimi, T., Tahriri, A., et al.: The Fundamental of Fabrication and Analytical Evaluation of a Sea Wave Impact Tester. J. Faculty of Eng. Tabriz University 31(2), 61–74 (2005)
6. Kostianoy, A.G., Kosarev, A.N., et al.: The Caspian Sea Environment: The Handbook of Environmental Chemistry /5. Springer, Heidelberg (2005)
7. Felezi, M.E., Nariman-zadeh, N., Darvizeh, A., Jamali, A., Teymoorzadeh, A.: A polynomial model for the level variations of Caspian Sea using Evolutionary Design of Generalized GMDH-type Neural Networks. WSEAS Transaction on Circuit and Systems 3(2) (2004) ISSN 1109-2734
8. Golub, G.H., Reinsch, C., et al.: Singular Value Decomposition and Least Squares Solutions. J. Numer. Math. 14(5), 403–420 (1970)

Appendix: List of Abbreviations

ANFIS..... Adaptive Neuro-Fuzzy Inference System
GA .....Genetic Algorithm
GMDH.....Group Method of Data Handeling
MSE.....Mean Square Error
SVD ..... Singular Value Decomposition
TSK..... Takagi-Sugeno-Kang

A Neuro-Fuzzy Control for TCP Network Congestion

S. Hadi Hosseini, Mahdieh Shabanian, and Babak N. Araabi

Abstract. We use Active Queue Management (AQM) strategy for congestion avoidance in Transmission Control Protocol (TCP) networks to regulate queue size close to a reference level. In this paper we present two efficient and new AQM systems as a queue controller. These methods are designed using Improved Neural Network (INN) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS). Our aim is low queue variation, low steady state error and fast response with using these methods in different conditions. Performance of the proposed controllers and disturbance rejection is compared with two well-known AQM methods, Adaptive Random Early Detection (ARED), and Proportional-Integral (PI). Our AQM methods are evaluated through simulation experiments using MATLAB.

1 Introduction

The congestion controllers strategies are used for prevent a starting congestion or recover from network congestion. Many of Transmission Control Protocol (TCP) schemes, adjusted window size for congestion avoidance, have been explored in the last two decades. The first used scheme, TCP Tahoe, and the next, TCP Reno [1].

Active Queue Management (AQM) scheme is a congestion avoidance strategy for TCP networks. Random Early Detection (RED) is a popular example of an AQM scheme [2]. Hollot and his colleagues in [3] have used the control theoretic approaches to determine the RED parameters. Several researchers have proposed as a controller for AQM system: the conventional controllers such as Proportional (P), Proportional-Integral (PI) [4], Proportional-Derivative (PD) [5], Proportional-Integral-Derivative (PID) [6], and adaptive controller such as Adaptive Random Early Detection (ARED) [7], and the heuristic methods such as fuzzy logic [8, 9, 10] and Neural Networks (NN) [11].

S. Hadi Hosseini · Mahdieh Shabanian
Science and Research branch, Islamic Azad University, Tehran, Iran
e-mail: {sh_hosseini,m_shabanian}@itrc.ac.ir

Babak N. Araabi
School of Electrical and Computer Eng., University of Tehran, Tehran, Iran
e-mail: araabi@ut.ac.ir

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 93–101.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

We present a more sophisticated adaptive control strategy for AQM in TCP networks using a dynamic Artificial Neural Network (ANN) AQM control. The neural network operates as an adaptive and robust [11]. TCP networks involve several stochastic variables with nonstationary time-varying statistics. Most of these factors are regarded as uncertainty in the AQM system. Thus, an AQM control requires adaptive stochastic control to overcome uncertainty and time-variance [17]. We choose a Multi Layer Perceptron (MLP) dynamic neural model. For simplicity, we derive a learning procedure by the gradient descent Back Propagation (BP) method. Then we improved this method with *Delta-Bar-Delta* algorithm [13] and designed a new Improved Neural Network (INN) AQM.

After the INN AQM, we choose Adaptive Neuro-Fuzzy Inference Systems (ANFIS) AQM, for TCP networks. This controller can promptly adapt its operation to the nonlinear time-varying and stochastic nature of TCP networks. As a result, unlike RED, classical linear control, and adaptive control approaches, ANFIS is able to determine satisfactory AQM system parameters values autonomously. Here we choose a four-input first order *Takagi-Sugeno* fuzzy model with three membership functions, in our five layer ANFIS architecture [12].

Performance of the proposed controllers is evaluated via simulations in MATLAB environment and compared with ARED and PI.

In the following section we present our improved neural network AQM TCP congestion control with adaptive *Delta-Bar-Delta* learning algorithm. ANFIS controller approach which is used in this paper is described in Sect. 3. In Sect. 4 simulation results and comparison between the proposed controllers with other controllers are given. Finally, the paper is concluded in Sect. 5.

2 Improved Neural Network AQM

The block diagram of TCP congestion control with the INN AQM proposed in this paper is shown in Fig. 1.

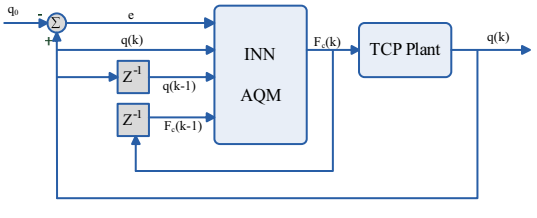


Fig. 1 INN AQM for TCP Congestion

In Fig. 1, q(k) is the queue size of available in router's buffer. The INN controller minimize the error signal (e), between the actual queue size q(k), and the reference queue target value q_0. The loss probability $P_d(k)$, is the control input to the TCP plant.

The MLP neural network model used in this paper is shown in Fig. 2.

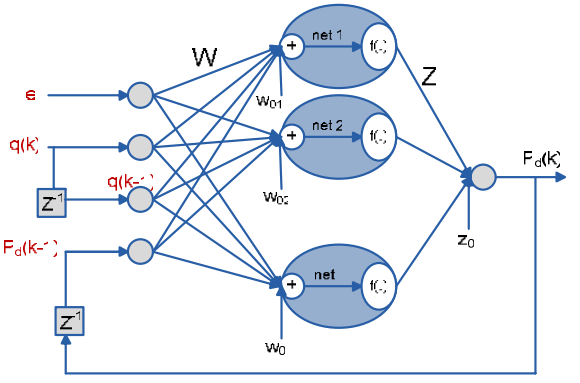


Fig. 2 MLP neural network for AQM

W_{ij} is the weight matrix of this model that number of neurons is i and number of input signals is j. We improve the neural network training algorithm and combine gradient descent BP algorithm and *Delta-Bar-Delta* algorithm (for adaptation of η) [13].

$$W_{ij} = W_{ij} - \eta_{ij} \frac{\partial J}{\partial W_{ij}} \tag{1}$$

$$\eta_{ij}(k+1) = \eta_{ij}(k) + \Delta\eta_{ij}(k) \tag{2}$$

$$\Psi(k) = \frac{\partial J}{\partial W_{ij}} \quad , \quad \overline{\Psi}(k) = (1-\varepsilon)\Psi(k) + \varepsilon\overline{\Psi}(k-1) \tag{3}$$

$$\Delta\eta_{ij}(k) = \begin{cases} K & \text{if} \quad \overline{\Psi}(k-1)\Psi(k) > 0 \\ -\beta\eta_{ij}(k) & \text{if} \quad \overline{\Psi}(k-1)\Psi(k) < 0 \\ 0 & \text{else} \end{cases} \tag{4}$$

Where J is our cost function and $K > 0$, $\beta \in [0,1]$, $\varepsilon \in [0,1]$ are constants. We show that the mentioned algorithm is better than previous methods in tracking TCP queue size.

3 ANFIS AQM

The block diagram of TCP congestion control with the ANFIS AQM proposed in this paper is similar to INN AQM block diagram. In this approach, the INN controller is replaced with the ANFIS controller. The ANFIS model used in this paper is shown in Fig. 3. Our ANFIS model is a simple *Takagi-Sugeno* fuzzy model [12]. It is includes four-inputs with three Gaussian membership functions for each input signals. Parameters in this layer are referred to as Premise Parameters, such as c_i and σ_i. The membership function is given by:

$$\mu(x) = \exp(\frac{-(x-c)^2}{2\sigma^2}) \tag{5}$$

This model includes five layers: Layer 1 consists of *Premise Parameters*, Layer 2 is a product layer, Layer 3 is a normalized layer, Layer 4 consists of *Consequent Parameters*, and Layer 5 is a summation layer. We have 3^4=81 rules. The *Consequent Parameters* are given by:

$$F_j = \alpha_{ij} u_i + \alpha_{1(j+1)} \tag{6}$$

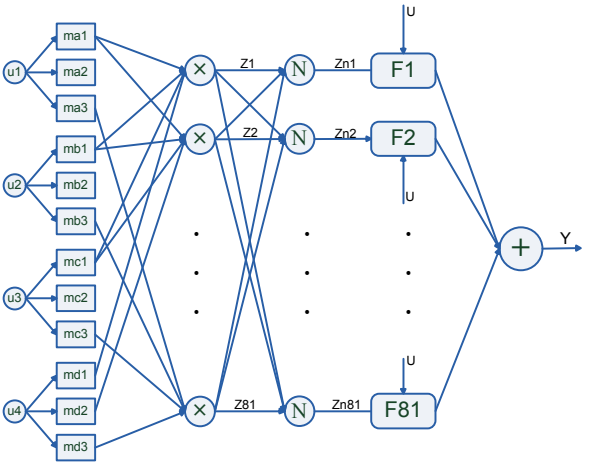


Fig. 3 ANFIS model for AQM

Where i is number of input signals, u is input signal, and j is number of rules. We fixed the *Premise Parameters* and membership functions cleverly that they shown in Fig. 4 and then trained the *Consequent Parameters* with gradient descent BP algorithm.

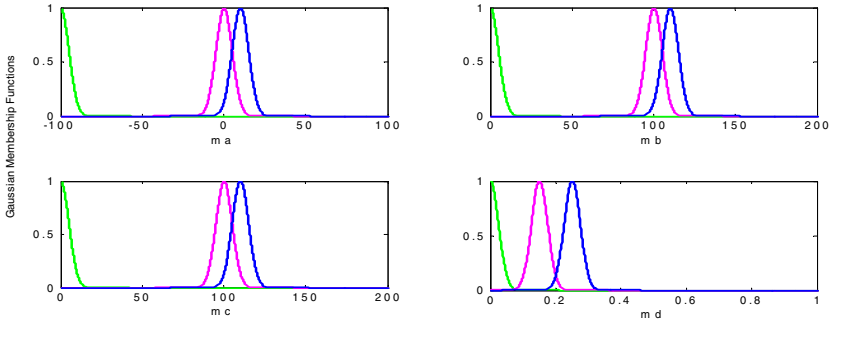


Fig. 4 ANFIS membership functions

4 Simulations and Results

We evaluate performance of the proposed INN and ANFIS AQM methods via simulations using MATLAB subroutine. To comparison the results, we also simulate the PI and ARED AQM schemes. The parameters of the PI controller are given in [4] and the parameters of the ARED are defined in [16]. PI parameters are: K_i=0.001, and K_p=0.0015, and also ARED parameters are: a=0.01, b=0.9, min_{th}=80, and max_{th}=159.

4.1 Simulation 1

In this simulation we use the nominal values known to controller parameters and compare four AQM algorithms: ANFIS, INN, PI and ARED. The scenario defined in [15] follows: N_n=50 (TCP sessions), C_n=300 (packet/sec), T_p=0.2 (sec), therefore R_{0n}=0.533 (sec) and W_{0n}=3.2 (packets). The desired queue lengths is q_0=100. Furthermore, propagation links delay, T_p is used as a random number. Simulation results are depicted in Fig. 5.

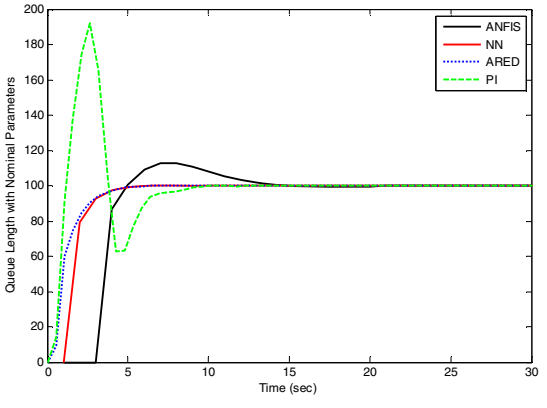


Fig. 5 Comparison of four AQM algorithms in nominal condition

In this case, the PI and ANFIS controller have an oscillatory behavior, but it is not important. The PI essentially is a fast controller with a high overshoot. The INN and ANFIS have a delay time because they are trained by their previous values of queue length. In the steady state, all of results are similar and good.

In Table 1 the mean and variance of queue length for four AQM are given. In the all of the simulations the mean and variance of queue computed in steady state condition from 15 sec to 100 sec.

4.2 Simulation 2

In this simulation we evaluate the robustness of the INN and ANFIS controller against variations in the network parameters. The numbers of sessions (N), link capacity (C) and propagation links delay (T) are changed during the simulation. First we consider the constant and real values for these parameters that given in [15]. This scenario follows: N_p=40 (TCP sessions), C_p=250 (packet/sec), and T_p = 0.3 (sec). These values are very different from the nominal values. After 20 seconds we increase those to twice of the previous values. Then after 20 seconds we decrease them to the nominal values at 40th second. Those parameters are decreased to half of the initial values at 60th second and returned to the initial values at 80th second. Simulation results are depicted in Fig. 6.

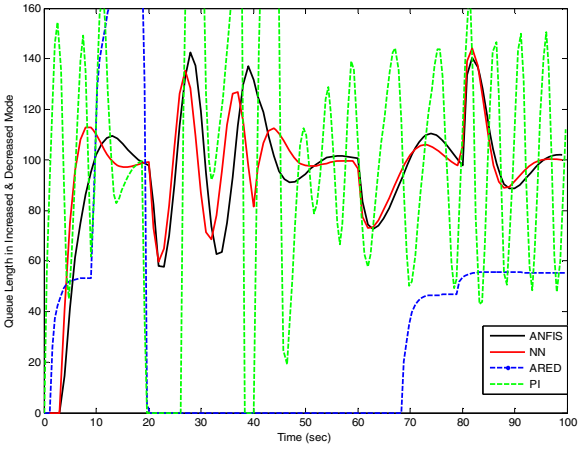


Fig. 6 Comparison of four AQM algorithms in real condition

As is shown in Fig. 6 and Table 1, queue length regulation using the ANFIS controller is considerably better than the others. Moreover, the ARED method could not track the desired queue and the PI method has an oscillatory treat in queue tracking. According to the Table 1, the ANFIS controller is better than the INN controller, because the variation of the ANFIS AQM is low.

4.3 Simulation 3

We evaluate the robustness of the proposed controller with respect to variations in the network parameters. The number of TCP flows N is considered as a normally distributed random signal with mean 45, and standard deviation 6, that it is added to a pulse train of period 50 (sec) and amplitude of 5. Moreover, the link capacity C is a normally distributed random signal with mean of 250 (packets/sec), and

standard deviation 6 (packets/sec), that it is added to a pulse of period 80 (sec) and amplitude of 50 (packets/sec). Also the propagation delay T_p is a normally distributed random signal with mean of 0.2 (sec), and standard deviation 2 (ms), that it is added to a pulse of period 50 (sec) and amplitude of 10 (ms). The sampling time is 0.53 (sec) for any parameters. These parameters are shown in Fig. 7.

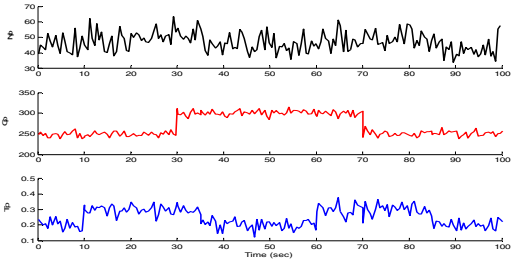


Fig. 7 Variation of N, C and T_p parameters corresponding to Simulation 3

Fig. 8 shows the queue regulation for four AQM methods. According to the Table 1 and Fig. 8, the proposed methods (INN and ANIS) are better than the others.

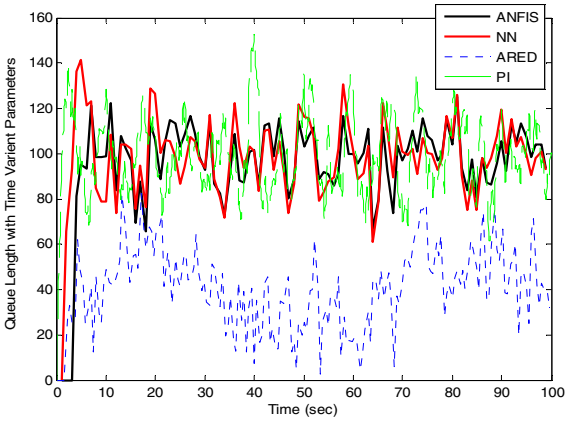


Fig. 8 Comparison of four AQM algorithms in time-variant condition

As is shown in Fig. 8, queue length regulation using the ANFIS controller is considerably better than the others. Moreover, the PI controller's queue length variation is much higher than the ANFIS controller. According to the Table 1, the ARED AQM's queue length mean is far from the desired value. In addition, the ANFIS controller is better than the INN controller, because the ANFIS AQM has a high mean of queue. Also, the queue length variance of ANFIS AQM has a low.

Table 1 Comparison of four AQM algorithms in three simulations

Simulation	AQM	Mean Queue Size (Packets)	Variance Queue Size (Packets)
1	ARED	100.05	0.001
	PI	99.99	0.001
	INN	100.02	2.84 * e-5
	ANFIS	100.00	1.26 * e-9
2	ARED	35.35	2600.35
	PI	99.42	2782.81
	INN	100.44	232.36
	ANFIS	100.07	183.55
3	ARED	41.51	302.68
	PI	99.82	1759.65
	INN	99.23	185.49
	ANFIS	99.58	136.54

5 Conclusions

We presented two novel AQM methodology using a dynamic neural network and neuro-fuzzy system for TCP congestion control. Both methods acted as a feedback controller to maintain the actual queue size close to a reference target. The neuro-fuzzy (ANFIS) AQM is trained by a gradient descent BP algorithm, but the improved neural network (INN) AQM is trained by a modified algorithm. In this modified algorithm we combined gradient descent BP algorithm with *Delta-Bar-Delta* algorithm. We applied proposed AQM systems to a single bottleneck network supporting multiple TCP flows. Three scenarios were examined in the simulation experiments to compare ANFIS and INN AQM to ARED and PI AQM.

The PI AQM resulted in queue saturation and larger overshoot. Also the result of ARED AQM is far from a reference target. However, ANFIS and INN AQM reduced overshoot and eliminated saturation and steady state error. ANFIS and INN AQM had the best regulation performance. Especially for real parameters (Sect. 4.2) and also for the case of time-varying TCP dynamics (Sect. 4.3), the ANFIS AQM was superior. We conclude that ANFIS AQM is an effective adaptive controller in TCP networks.

Future work will extend our results to more complex network scenarios, such as short TCP connections or noise disturbance networks, and will include various simulation scenarios using a network simulation tool such as NS-2 and OPNET to verify our results.

Acknowledgments

The authors sincerely thank the Iran Telecommunication Research Center (ITRC) for their financial support under grant number 11547.

References

[1] Jacobson, V.: Congestion avoidance and control. In: Proc. of SIGCOMM 1988, pp. 314–329 (1988)

[2] Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ACM Trans. on Networking 1, 397–413 (1993)

[3] Hollot, C.V., Misra, V., Towsley, D., Gong, W.B.: A Control Theoretic Analysis of RED. In: Proc. of IEEE INFOCOM, pp. 1510–1519 (2001)

[4] Hollot, C.V., Misra, V., Towsley, D., Gong, W.B.: Analysis and design of controllers for AQM routers supporting TCP flows. IEEE Trans. on Automatic Control 47, 945–959 (2002)

[5] Sun, C., Ko, K.T., Chen, G., Chen, S., Zukerman, M.: PD-RED: To improve the performance of RED. IEEE Communication Letters 7, 406–408 (2003)

[6] Ryu, S., Rump, C., Qiao, C.: A Predictive and robust active queue management for Internet congestion control. In: Proc. of ISCC 2003, pp. 1530–1346 (2003)

[7] Zhang, H., Hollot, C.V., Towsley, D., Misra, V.: A self-tuning structure for adaptation in TCP/AQM networks. In: Proc. of IEEE/GLOBECOM 2003, vol. 7, pp. 3641–3646 (2003)

[8] Hadjadj, Y., Nafaa, A., Negru, D., Mehaoua, A.: FAFC: Fast Adaptive Fuzzy AQM Controller for TCP/IP Networks. IEEE Trans. on Global Telecommunications Conference 3, 1319–1323 (2004)

[9] Taghavi, S., Yaghmaee, M.H.: Fuzzy Green: A Modified TCP Equation-Based Active Queue Management Using Fuzzy Logic Approach. In: Proc. of IJCSNS, vol. 6, pp. 50–58 (2006)

[10] Hadjadj, Y., Mehaoua, A., Skianis, C.: A fuzzy logic-based AQM for real-time traffic over internet. Proc. Computer Networks 51, 4617–4633 (2007)

[11] Cho, H.C., Fadali, M.S., Lee, H.: Neural Network Control for TCP Network Congestion. In: Proc. American Control Conference, vol. 5, pp. 3480–3485 (2005)

[12] Jang, J.R., Sun, C., Mizutani, E.: Neuro-Fuzzy and Soft Computing. Prentice-Hall, Englewood Cliffs (1997)

[13] Haykin, S.: Neural Networks: A comprehensive foundation. Prentice Hall, Englewood Cliffs (1999)

[14] Misra, V., Gong, W.B., Towsley, D.: Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In: Proc. of ACM/SIGCOMM, pp. 151–160 (2000)

[15] Quet, P.F., Ozbay, H.: On the design of AQM supporting TCP flows using robust control theory. IEEE Trans. on Automatic Control 49, 1031–1036 (2004)

[16] Floyed, S., Gummadi, R., Shenker, S.: Adaptive RED: An Algorithm for Increasing the Robustness of RED s Active Queue Management. Technical Report, ICSI (2001)

[17] Cho, H.C., Fadali, S.M., Lee, H.: Adaptive neural queue management for TCP networks. In: Proc. Computers and Electrical Engineering, vol. 34, pp. 447–469 (2008)

Use of Remote Sensing Technology for GIS Based Landslide Hazard Mapping

S. Prabu, S.S. Ramakrishnan, Hema A. Murthy, and R.Vidhya

Abstract. This purpose of this study is a combined use of socio economic, remote sensing and GIS data for developing a technique for landslide susceptibility mapping using artificial neural networks and then to apply the technique to the selected study areas at Nilgiris district in Tamil Nadu and to analyze the socio economic impact in the landslide locations. Landslide locations are identified by interpreting the satellite images and field survey data, and a spatial database of the topography, soil, forest, and land use. Then the landslide-related factors are extracted from the spatial database. These factors are then used with an artificial neural network (ANN) to analyze landslide susceptibility. Each factor's weight is determined by the back-propagation training method. Different training sets will be identified and applied to analyze and verify the effect of training. The landslide susceptibility index will be calculated by back propagation method and the susceptibility map will be created with a GIS program. The results of the landslide susceptibility analysis are verified using landslide location data. In this research GIS is used to analysis the vast amount of data very efficiently and an ANN to be an effective tool to maintain precision and accuracy. Finally the artificial neural network will prove it's an effective tool for

S. Prabu
Research Fellow, Institute of Remote Sensing, College of Engineering Guindy,
Anna University, Chennai, Tamil Nadu, India
e-mail: sevu_prabu@yahoo.co.in

S.S. Ramakrishnan
Professor, Institute of Remote Sensing, College of Engineering Guindy,
Anna University, Chennai, Tamil Nadu, India
e-mail: ssramki@annauniv.edu

Hema A. Murthy
Professor, Department of Computer Science and Engineering,
Indian Institute of Technology Madras, Chennai, Tamil Nadu, India
e-mail: hema@lantana.tenet.res.in

R.Vidhya
Assistant Professor, Institute of Remote Sensing, College of Engineering Guindy,
Anna University, Chennai, Tamil Nadu, India
e-mail: rvidhya@annauniv.edu

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 103–113.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

analyzing landslide susceptibility compared to the conventional method of landslide mapping. The Socio economic impact is analyzed by the questionnaire method. Direct survey has been conducted with the people living in the landslide locations through different set of questions. This factor is also used as one of the landslide causing factor for preparation of landslide hazard map.

1 Introduction

Landslide risk is defined as the expected number of lives lost, persons injured, damage to property and disruption of economic activity due to a particular landslide hazard for a given area and reference period (Varnes, 1984)[10]. When dealing with physical losses, (specific) risk can be quantified as the product of vulnerability, cost or amount of the elements at risk and the probability of occurrence of the event. When we look at the total risk, the hazard is multiplied with the expected losses for all different types of elements at risk (= vulnerability * amount), and this is done for all hazard types. Schematically, this can be represented by the following formula (1):

$$\text{Risk} = \Sigma\,(H * \Sigma\,(V * A)) \tag{1}$$

Where:

H = Hazard expressed as probability of occurrence within a reference period (e.g., year)

V = Physical vulnerability of a particular type of element at risk (from 0 to 1)

A = Amount or cost of the particular elements at risk (e.g., number of buildings, cost of buildings, number of people, etc.). Theoretically, the formula would result in a so-called risk curve, containing the relation between all events with different probabilities, and the corresponding losses.

Out of the factors mentioned in the formula for risk assessment, the hazard component is by far the most difficult to assess, due to the absence of a clear magnitude-frequency relation at a particular location, although such relations can be made over larger areas. Furthermore, the estimation of both magnitude and probability of landsliding requires a large amount of information on the following aspects:

- Surface topography
- Subsurface stratigraphy
- Subsurface water levels, and their variation in time
- Shear strength of materials through which the failure surface may pass Unit weight of the materials overlying potential failure planes and the intensity and probability of triggering factors, such as rainfall and earthquakes.

All of these factors, required to calculate the stability of individual slopes, have a large spatial variation, and are only partly known, at best. If all these factors would be known in detail it would be possible to determine which slopes would generate landslides of specific volumes and with specific run out zones for a given period of time.

1.1 Digital Techniques for Landslide Change Detection

Despite the theoretical availability of high resolution satellite images, aerial photographs are used more extensively for landslide studies because they have been in existence for a long time and have a suitable spatial resolution. Techniques for change detection using digital aerial photos are often based on the generation of high accurate orthophotos, using high precision GPS control points, for images from different periods. A detailed procedure is given in Casson *et al.* (2003)[3] with a multi-temporal example from the La Clapiere landslide in France. Hervas *et al.* (2003)[5], and Van Westen and Lulie (2003)[9] have made similar attempts for the Tessina landslide in Italy.

1.2 GIS Data Analysis and Modeling for Landslide Risk Assessment

The number of recent publications on various methods for GIS based landslide hazard assessment is overwhelming, especially when compared with those that also deal with landslide vulnerability and risk assessment, which are still very few. Overviews and classification of GIS based landslide hazard assessment methods can be found in Soeters and Van Westen (2003)[9], Leroi (1996)[6], Carrara *et al.* (1995[1], 1999)[2], and Van Westen (2000)[8]. In terms of software used, GIS systems such as ArcInfo, ArcView, ArcGIS, SPANS, IDRISI, GRASS and ILWIS are mostly used and statistical packages such as Statgraph or SPSS. Most GIS systems are good in data entry, conversion, management, overlaying and visualization, but not very suitable for implementing complex dynamic simulation models. Some GIS systems are specifically designed for implementing such dynamic models (PCRaster, 2000)[7].

1.2.1 Landslide Risk Analysis

Risk is the result of the product of probability (of occurrence of a landslide with a given magnitude), costs (of the elements at risk) and vulnerability (the degree of damage of the elements at risk due to the occurrence of a landslide with a given magnitude). A complete risk assessment involves the quantification of a number of different types of losses (FEMA, 2004)[4], such as:

- *Losses associated with general building stock:* structural and nonstructural cost of repair or replacement, loss of contents.
- *Social losses:* number of displaced households; number of people requiring temporary shelter; casualties in four categories of severity (based on different times of day)
- *Transportation and utility lifelines*: for components of the lifeline systems: damage probabilities, cost of repair or replacement and expected functionality for various times following the disaster.
- *Essential facilities*: damage probabilities, probability of functionality, loss of beds in hospitals.

- *Indirect economic impact*: business inventory loss, relocation costs, business income loss, employee wage loss, loss of rental income, long-term economic effects on the region.

The quantification of landslide risk is often a difficult task, as both the landslide intensity and frequency will be difficult to calculate for an entire area, even with sophisticated methods in GIS. In practice, often simplified qualitative procedures are used, such as the neural network model and analytical hierarchic process.

2 Study Area

Study area is geographically located between 76° 14. 00. and 77° 02. 00. E longitudes and 11° 10. 00. and 11° 42. 00. N latitudes. The Nilgiris district is a mountainous terrain in the North West part of Tamil Nadu, India.

3 Data Requirements

Data	GIS data type
Landslide (1:5,000)	ARC/INFO polygon coverage
Geological map (1:50,000)	ARC/INFO polygon coverage
Landuse map (1:50,000)	ARC/INFO grid
Rainfall map (1:50,000)	ARC/INFO polygon coverage
Slope map (1:50,000)	ARC/INFO polygon coverage
Soil map(1:50,000)	ARC/INFO Polygon Coverage

4 Construction of Spatial Database Using GIS

To apply the artificial neural network, a spatial database is created that took landslide-related factors such as topography, soil, forest, and land use into consideration. Landslide occurrence areas are detected from both Indian Remote Sensing (IRS) and field survey data. In the study area, rainfall triggered debris flows and shallow soil slides are the most abundant. Maps relevant to landslide occurrence are constructed in a vector format spatial database using the GIS ARC/INFO or Arc Map software package.

These included 1:50000 scale topographic maps, 1:50,000 scale soil maps, and 1:50,000 scale forest maps. Contour and survey base points that had an elevation value read from a topographic map are extracted, and a digital elevation model (DEM) is constructed. The DEM has a 10 m resolution and will be used to calculate the slope, aspect, and curvature. Drainage and topographic type will be extracted from the soil database. Forest type, timber age, timber diameter, and timber density will be extracted from forest maps. Land use was classified from Landsat TM satellite imagery.

Both the calculated and extracted factors are converted to form a 10×10 m^2 grid (ARC/INFO grid type), and then it will be converted to ASCII data for use with the artificial neural network program. Then the back propagation network

(BPN) is used for training the network by adjusting the weights between the nodes. After the training the BPN was used as the feed forward network for the new area classification and the verification has been done by comparing with the field verification and map data. This is shown in fig.1.

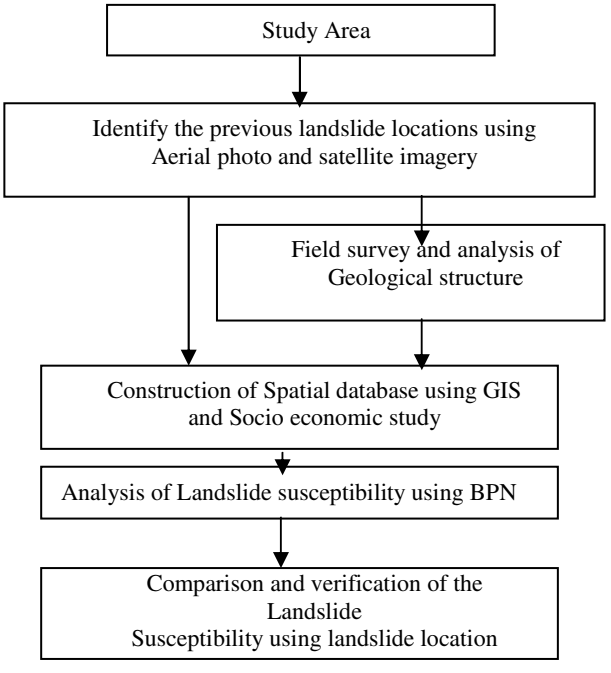


Fig. 1 Flowchart of Methodology

5 The Artificial Neural Network

An artificial neural network is a "computational mechanism able to acquire, represent, and compute a mapping from one multivariate space of information to another, given a set of data representing that mapping". The back propagation training algorithm (BPN) is the most frequently used neural network method and is the method used in this study. The back-propagation training algorithm is trained using a set of examples of associated input and output values. The purpose of an artificial neural network is to build a model of the data-generating process, so that the network can generalize and predict outputs from inputs that it has not previously seen.

There are two stages involved in using neural networks for multi-source classification: the training stage, in which the internal weights are adjusted; and the classifying stage. Typically, the back-propagation algorithm trains the network until some targeted minimal error is achieved between the desired and actual

output values of the network. Once the training is complete, the network is used as a feed-forward structure to produce a classification for the entire data.

A neural network consists of a number of interconnected nodes. Each node is a simple processing element that responds to the weighted inputs it receives from other nodes. The arrangement of the nodes is referred to as the network architecture (Fig. 2).

The receiving node sums the weighted signals from all the nodes that it is connected to in the preceding layer. Formally, the input that a single node receives is weighted according to Equation (2).

$$net_j = \Sigma W_{ij}\, O_i \tag{2}$$

Where W_{ij} represents the weights between nodes i and j, and o_i is the output from node j, given by

$$O_j = f\,(net_j) \tag{3}$$

The function f is usually a nonlinear sigmoid function that is applied to the weighted sum of inputs before the signal propagates to the next layer. One advantage of a sigmoid function is that its derivative can be expressed in terms of the function itself:

$$f'(net_j)=f(net_j)*(1-f(net_j)) \tag{4}$$

The network used in this study consisted of three layers. The first layer is the input layer, where the nodes are the elements of a feature vector. The second layer is the internal or "hidden" layer. The third layer is the output layer that presents the output data. Each node in the hidden layer is interconnected to nodes in both the preceding and following layers by weighted connections.

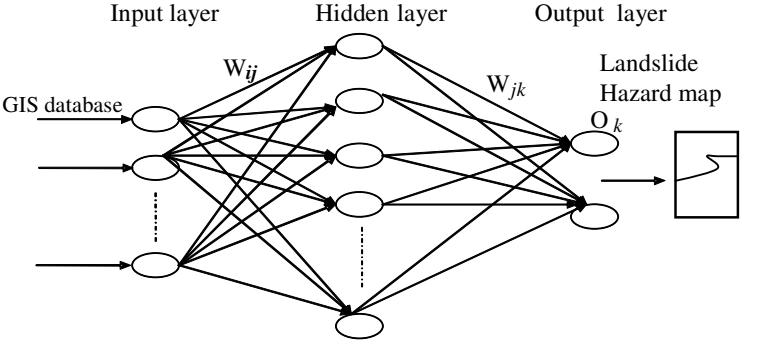


Fig. 2 Architecture of artificial neural network

The error, E, for an input training pattern, t, is a function of the desired output vector, d, and the actual output vector, o, given by

$$E=1/2\sum (d_k-o_k) \tag{5}$$

The error is propagated back through the neural network and is minimized by adjusting the weights between layers. The weight adjustment is expressed as

$$W_{ij}\,(n+1) = \eta\,(\delta_j o_i) + \alpha\, W_{ij} \tag{6}$$

Where η is the learning rate parameter (set to $\eta = 0.01$ in this study), δ_j is an index of the rate of change of the error, and α is the momentum parameter.

The factor δ_j is dependent on the layer type. For example,

For hidden layers, $$\delta_j = (\sum \delta_k\, w_{jk})\, f'\,(net_j) \tag{7}$$

and for output layers, $$\delta_j = (d_k - o_k))\, f'\,(net_k) \tag{8}$$

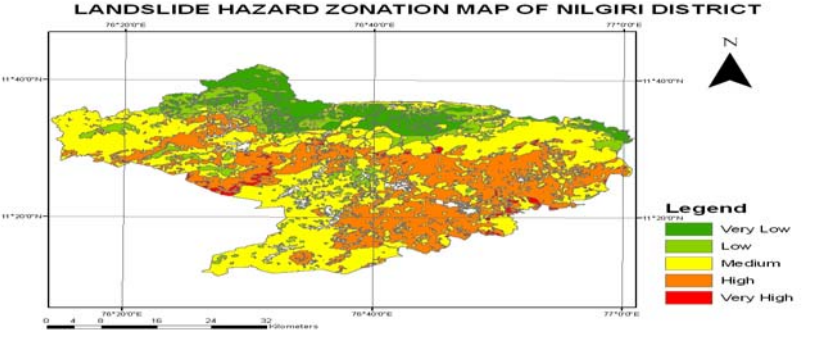


Fig. 3 Landslide susceptibility map prepared using neural network

This process of feeding forward signals and back-propagating the error is repeated iteratively until the error of the network as a whole is minimized or reaches an acceptable magnitude. Using the back-propagation training algorithm, the weights of each factor can be determined and may be used for classification of data (input vectors) that the network has not seen before. From Equation (3), the effect of an output, o_j, from a hidden layer node, j, on the output, o_j, from an output layer (node k) can be represented by the partial derivative of o_k with respect to o_j as

$$\partial o_k / \partial\, o_j = f\,'(net_k) \times \partial\,(net_k)/\partial o_j = f\,'(net_k) \times W_{jk} \tag{9}$$

Equation (9) produces both positive and negative values. If the effect's magnitude is all that is of interest, then the importance (weight) of node j relative to another node O_J in the hidden layer may be calculated as the ratio of the absolute values derived.

6 Landslide Susceptibility Forecast Mapping and Verification

The calculated landslide susceptibility index values computed using back propagation is converted into an ARC/INFO grid. Then a landslide susceptibility

map is created. The final landslide susceptibility maps are prepared shown in Fig.3. Verification is performed by comparing existing landslide data with the landslide susceptibility analysis results of the study area.

7 The Analytical Hierarchic Process

The Analytical Hierarchy Process (AHP), a theory for dealing with complex, technological, economical, and socio-political problems is an appropriate method for deriving the weight assigned to each factor. Basically, AHP is a multi-objective, multi-criteria decision-making approach to arrive at a scale of preference among a set of alternatives. AHP gained wide application in site selection, suitability analysis, regional planning, and landslide susceptibility analysis.

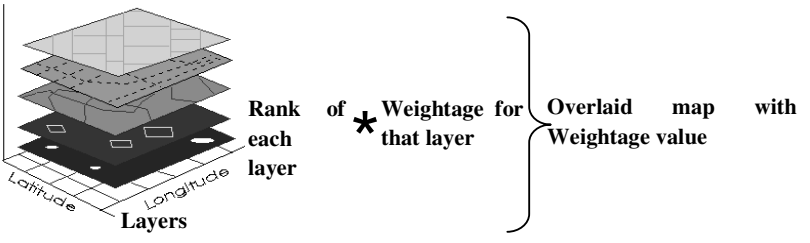


Fig. 4 Weighted overlay

The AHP is employed to determine the effect of the data in this database in producing the landslide susceptibility map. With this method, the effect of the subgroups of the data layer and the effect value related to each other are quantitatively determined. It has been shown that the use of the AHP method produces a practical and realistic result to define the factor weights in the landslide susceptibility model. (Fig. 4).

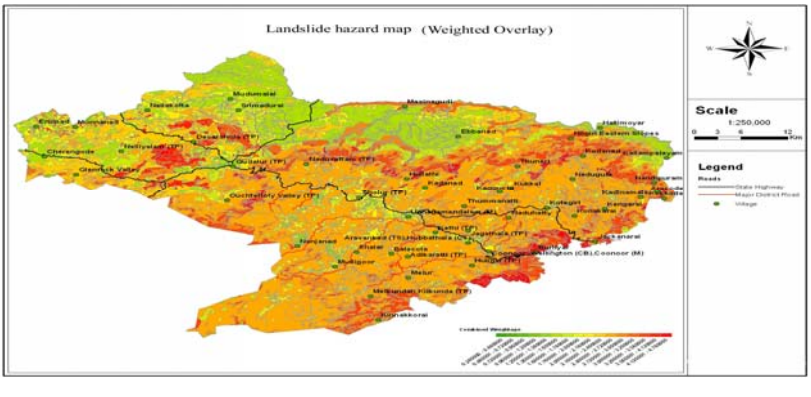


Fig. 5 Landslide hazard mapping through AHP (Weighted Overlay)

The table 1 shows that the ranks assigned to each layer based on their contribution to cause the landslide and weightages also assigned to each layer. Fig.4 shows the final landslide hazard map prepared through Analytical Hierarchic Process method. This map prepared based on the scores of each landslide causing factor multiplied with the weightages given for each factor based on contribution for causing landslide. In our chosen study area only three types of geology has been found so the ranks has been assigned starting from 2 to 4 and rainfall is the major factor for causing landslide in our area so the ranks have been assigned as 4 (rainfall from 1000 to 1600mm) and 5 (1600 to 2800mm).

The AHP allows the consideration of both objective and subjective factors in selecting the best alternative. Despite the widespread use of the AHP in diverse

Table. 1 Weightages and Ranks assigned for each layers based on the contribution for the Landslide Cause

Layer	Weightage	Class	Rank in 5 point Scale
Slope	40	0-8%	1
		8-15%	2
		15-30%	3
		30-60%	4
		>60%	5
Land use	36	Arable	4
		Forest	1
		Scrub and Grass	5
		Water Body	0
		Builtup land	0
Rainfall	12	1000-1200	4
		1200-1400	4
		1400-1600	4
		1600-2000	5
		2000-2400	5
		2400-2800	5
Geology	12	Ultramafic rocks (Mylonite)	2
		Fuchsite quartzite, schistose quartzite, sillimani	2
		Fissile hornblende biotite gneiss	2
		Hornblende-biotite gneiss	3
		Garnetiferous quartofeldspathic gneiss	3
		Felsite	3
		Metagabbro, Pyroxenite, Pyroxene granulite	4
		Charnockite	4

decision problems, this multi-attribute approach has not been without criticisms. However, in fact, it is one of the most popular multi-criteria decision-making methodologies available today. Thus, the areas in which the AHP is applied are diverse and numerous. The popularity of the AHP is due to its simplicity, flexibility, ease of use and interpretation, etc. in analyzing complex decision problems.

8 Conclusion

Landslides are one of the most hazardous natural disasters, not only in India, but around the world. Government and research institutions worldwide have attempted for years to assess landslide hazards and their associated risks and to show their spatial distribution. An artificial neural network approach was used to estimate areas susceptible to landslides using a spatial database constructed through GIS for a selected study area.

In this neural network method, it is difficult to follow the internal processes of the procedure. There is a need to convert the database to another format, such as ASCII, the method requires data be converted to ASCII for use in the artificial neural network program and later reconverted to incorporate it into a GIS layer. Moreover, the large amount of data in the numerous layers in the target area cannot be processed in artificial neural network programs quickly and easily. Using the forecast data, landslide occurrence potential can be assessed, but the landslide events cannot be predicted. The socio economic survey is used as one factor for this mapping will be an added advantage for producing accurate result with neural network method. Finally, the neural network has proved as an efficient method for landslide mapping compared with other methods like analytical hierarchic process.

Acknowledgments. The authors would like to thank the referees and the researchers of WSC 2008 for their accurate review of the manuscript and their valuable comments.

References

[1] Carrara, A., Cardinali, M., Guzzetti, F., Reichenbach, P.: GIS-based techniques for mapping landslide hazard. In: Carrara, A., Guzzetti, F. (eds.) Geographical Information Systems in Assessing Natural Hazards, pp. 135–176. Kluwer Publications, Dordrecht (1995)

[2] Carrara, A., Guzzetti, F., Cardinali, M., Reichenbach, P.: Use of GIS technology in the prediction and monitoring of landslide hazard. Natural Hazards 20(2-3), 117–135 (1999)

[3] Casson, B., Delacourt, C., Baratoux, D., Allemand: seventeen years of the " La Clapiere" landslide evolution analysed frpm prtho-rectified aerial photographs. Engineering Geology 68(1-2), 123–139 (2003)

[4] FEMA. Federal Emergency Management Agency: HAZUS-MH. Software tool for loss estimation (2004), `http://www.fema.gov/hazus/index.shtm` (Verified: 3/3/2004)

[5] Hervás, J., Barredo, J.I., Rosin, P.L., Pasuto, A., Mantovani, F., Silvano, S.: Monitoring landslides from optical remotely sensed imagery: the case history of Tessina landslide, Italy. Geomorphology 54(1-2), 63–75 (2003)
[6] Leroi, E.: Landslide hazard - Risk maps at different scales: Objectives, tools and development. In: Senneset, K. (ed.) Landslides – Glissements de Terrain, Balkema, Rotterdam, pp. 35–51 (1996)
[7] PCRaster, PCRaster Environmental Software, Manual version 2, Faculty of Geographical Sciences, Utrecht University, 367 p. (2000)
[8] Van Westen, C.J.: The modeling of landslide hazards using GIS. Surveys in Geophysics 21(2-3), 241–255 (2000)
[9] Van Westen, C.J., Lulie Getahun, F.: Analyzing the evolution of the Tessina landslide using aerial photographs and digital elevation models. Geomorphology 54(1-2), 77–89 (2003)
[10] Varnes, D.J.: Landslide Hazard Zonation: A Review of Principles and Practice. United Nations International, Paris (1984)

An Analysis of the Disturbance on TCP Network Congestion

Mahdieh Shabanian, S. Hadi Hosseini, and Babak N. Araabi

Abstract. In this study, the disturbance and uncertainty on nonlinear and time varying systems as Active Queue Management (AQM) is analyzed. Many of AQM schemes have been proposed to regulate a queue size close to a reference level with the least variance. We apply a normal range of disturbances and uncertainty such as variable user numbers, variable link capacity, noise, and unresponsive flows, to the three AQM methods: Random Early Detection (RED), Proportional-Integral (PI) and Improved Neural Network (INN) AQM. Then we examine some important factors for TCP network congestion control such as queue size, drop probability, variance and throughput in *NS-2* simulator, and then compare three AQM algorithms with these factors on congestion conditions. We present the performance of the INN controller in desired queue tracking and disturbance rejection is high.

1 Introduction

Congestion in Transmission Control Protocol (TCP) networks is the result of high needs for limited network resources. Moreover, when any high-speed links receive to one low-speed link, the congestion occurs. If the congestion continues, the undesired collapse phenomenon will occur. Active Queue Management (AQM) schemes are strategies which are implemented in routers to moderate TCP (Transmission Control Protocol) traffic. Random Early Detection (RED) is a popular method of an AQM scheme that presented by Floyd, and Jacobson in 1993 [2].

Although, this AQM is very simple and useful, however dynamics of the TCP networks are time-variant, and it is difficult to design RED parameters in order to obtain good performance under different congestion scenarios. In addition, it is difficult when we have any disturbance in TCP networks.

Using the control theory, conventional controllers such as Proportional (P), Proportional-Integral (PI) [4], Proportional-Derivative (PD) [5], Proportional-Integral-Derivative (PID) [6], and adaptive controller such as Adaptive Random

Mahdieh Shabanian · S. Hadi Hosseini
Science and Research branch, Islamic Azad University, Tehran, Iran
e-mail: (m_shabanian, sh_hosseini)@itrc.ac.ir

Babak N. Araabi
School of Electrical and Computer Eng., University of Tehran, Tehran, Iran
e-mail: araabi@ut.ac.ir

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 115–123.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

Early Detection (ARED) [7], have been designed as AQM methods. In practice, we show that, the choice of control parameters in these methods like RED is very difficult due to system uncertainty.

Moreover, when any disturbance is combined with TCP network, setting the parameters is very difficult and often is impossible. Therefore, parameter values must be adjusted to adapt to operational changes. But, intelligent and heuristic adaptation methods such as fuzzy logic [8], [9], [10], and neural networks [11] is better than classic methods when there is disturbance in the system. In addition, most control-theoretic AQM designed for time-varying stochastic systems.

The neural network controller can promptly adapt its operation to the nonlinear time-varying and stochastic nature of TCP networks. We consider a Multi Layer Perceptron (MLP) dynamic neural model because of its well-known advantages. For simplicity, a learning procedure by the gradient descent back-propagation (BP) method is derived [12]. Then for high adaptation against uncertainty and disturbance, we use the Improved Neural Network (INN) AQM [1].

In this study, a normal range of disturbance and uncertainty in TCP networks are explained and analyzed. The TCP system model of [14] is used. Performance of the proposed controller is evaluated via simulations in *NS-2* environment. The advantages of our proposed method are illustrated with compared to RED and PI, the two common AQM methods.

In the following section the dynamics of TCP/AQM networks in congestion avoidance mode is described. In Sect. 3, we describe INN AQM TCP congestion control with Adaptive *Delta-Bar-Delta* learning algorithm for this model. Simulation results and comparison between the effectiveness of the proposed controller with other controllers are given in Sect. 4. Finally, the paper is concluded in Sect. 5.

2 Dynamics of TCP/AQM Networks

A mathematical model of TCP that is developed in [3] using fluid-flow and stochastic differential equation is considered. A simplified version of the model is used, which ignores the TCP timeout mechanism, and it is described with (1) and (2).

$$\dot{W} = \frac{1}{R(t)} - \frac{W(t)}{2}\frac{W(t-R(t))}{R(t-R(t))}P_d(t-R(t)),\quad W \geq 0 \tag{1}$$

$$\dot{W} = \frac{1}{R(t)} - \frac{W(t)}{2}\frac{W(t-R(t))}{R(t-R(t))}P_d(t-R(t)),\quad W \geq 0 \tag{2}$$

$$\dot{q} = -C + \frac{N(t)}{R(t)}W(t),\quad q \geq 0 \tag{3}$$

Where W is the average TCP window size (packets) and q is the average queue length (packets). Both of them are positive and bounded quantities; i.e., $W \in [o, \overline{W}]$ and $q \in [0, \overline{q}]$ where $\overline{W}$ and $\overline{q}$ denote maximum window size and buffer capacity respectively. Also N, C, T_p and $R(t)=q(t)/C + T_p$ are the load factor (number of TCP sessions), link capacity (packets/sec), propagation delay (sec) and

round-trip time (sec) respectively. P_d is the probability of packet marking or dropping due to AQM mechanism at the router and takes value in [0,1] . In defined differential equations, (1) describes the TCP window control dynamic and (2) models the bottleneck queue length [3]. The 1/R(t) term in the right hand side of (1) models the window's additive increase and the term W(t)/2 models the window's multiplicative decrease in response to packet marking. Queue length in (2) is modeled as difference between packet arrival rate N/R(t) and link capacity C. To design the controller (AQM), the small signal linearized model of these nonlinear dynamics is used. Assuming that the number of TCP session and link capacity is fixed and ignoring the dependence of time-delay argument t-R(t) on queue length, the perturbed variables about the operating point satisfy [3]:

$$\delta\dot{W}(t) = -\frac{N}{R_0^2 C}(\delta W(t) - \delta W(t - R_0)) - \frac{1}{R_0^2 C}(\delta q(t) - \delta q(t - R_0)) - \frac{R_0 C^2}{2N^2}\delta P_d(t - R_0) \tag{4}$$

$$\delta\dot{q}(t) = \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t) \tag{5}$$

3 Improved Neural Network AQM

The block diagram of TCP congestion control with the INN AQM proposed is shown in Fig. 1 [1].

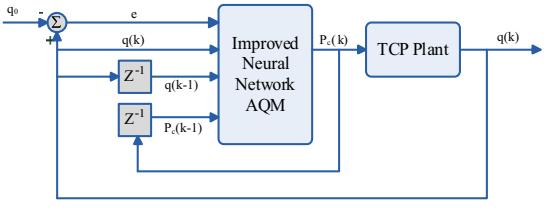


Fig. 1 INN AQM for TCP Congestion [1]

Our INN model includes three-layer perceptron with four input signals. The input vector of this neural network includes the error signal (e), queue size (q), as a feedback signal from system output, and the probability (P_d), as a feedback signal from neural models output.

The weight matrix of this model is W_{ij} that i is the number of neurons and j is the number of input signals. We improve the neural network training algorithm and combine gradient descent BP algorithm and *Delta-Bar-Delta* algorithm for adaptation of η [1].

$$W_{ij} = W_{ij} - \eta_{ij}\frac{\partial J}{\partial W_{ij}} \tag{6}$$

$$\eta_{ij}(k+1) = \eta_{ij}(k) + \Delta\eta_{ij}(k) \tag{7}$$

$$\Psi(k) = \frac{\partial J}{\partial W_{ij}} \quad , \quad \overline{\Psi}(k) = (1-\varepsilon)\Psi(k) + \varepsilon\overline{\Psi}(k-1) \tag{8}$$

$$\Delta\eta_{ij}(k) = \begin{cases} K & \text{if} \quad \overline{\Psi}(k-1)\Psi(k) > 0 \\ -\beta\eta_{ij}(k) & \text{if} \quad \overline{\Psi}(k-1)\Psi(k) < 0 \\ 0 & \text{else} \end{cases} \tag{9}$$

Where J is our cost function and $K > 0$, $\beta \in [0,1]$, $\varepsilon \in [0,1]$ are constants. We show that the mentioned algorithm is better than previous methods in tracking TCP queue size.

4 Simulations and Results

We evaluate performance of the proposed INN AQM method via simulations using Network Simulator (*NS-2*). *Dumbbell* model for TCP Network is used as benchmark. In Fig. 2 a simple bottleneck link between two routers and numerous TCP flows is shown.

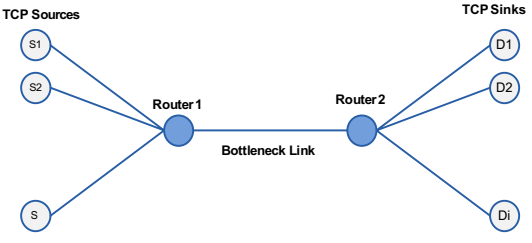


Fig. 2 *Dumbbell* model for TCP network

The PI and RED AQM schemes are simulated and their results are compared. The parameters of the PI controller are given in [4] and the parameters of the RED are defined in [13]. PI parameters are: K_i=0.001, K_p=0.0015 and RED parameters are: ω_q=0.0001, p_{max}=0.1, min_{th}=50, and max_{th}=150.

4.1 Simulation 1

In this simulation, we use the nominal values known to controller parameters and compare three AQM Algorithms: INN, PI and RED. The scenario defined in [13] is as follows: N_n=50 TCP sessions (S_i=D_i=N_n), C_n=300 (packet/sec), T_p=0.2 (sec), therefore R_{0n}=0.533 (sec) and W_{0n}=3.2 (packets). The desired queue lengths is q_0=100. Furthermore, the propagation links delay, T_p is used as a random number. In this case there is not any disturbance in the system. Simulation results are depicted in Fig. 3.

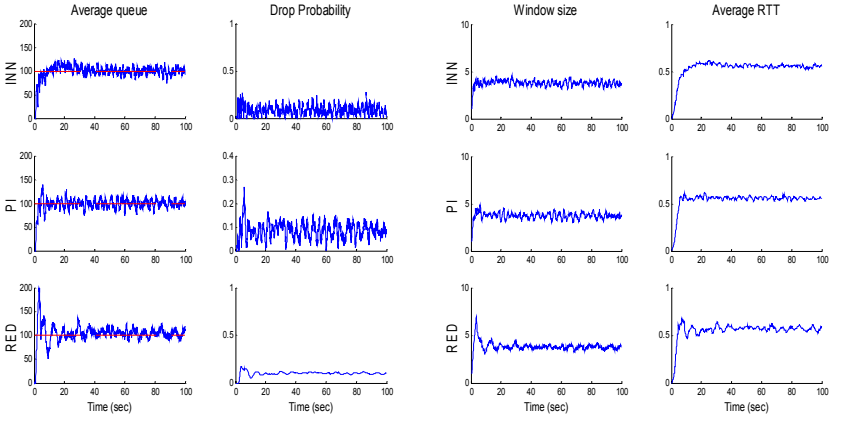


Fig. 3 Comparison of three AQM algorithms in nominal condition

There is a direct relation between variation of queue lengths and congestion window size. Therefore, the reason of decreasing the variation of RTT is the low queue lengths variation around q_0=100. In the PI controller, because of existence the integrator term in its structure, the amount of steady state error is zero. Also, the PI AQM has a fast response than the others. The PI controller has a minimum gain margin and phase margin.

The RED algorithm has a low drop probability; because in this method, the queue size is regulated in higher value than the desired one and the packets are remained in queue buffer for long times. Moreover, the RED algorithm response is slow because it uses average function for computation the probability of the drop. In general, the aim of congestion control algorithm is regulation of queue length with the least variance. The INN controller regulates the queue response without overshoot and with the least variance. Furthermore, average drop probability in the INN controller is smaller than the PI.

In Table 1, we show the average queue size, standard deviation (STD) queue, loss rate and throughput for three AQM in nominal condition that is result of simulation 1.

Table 1 Compare three AQM algorithms in nominal condition

	INN	PI	RED
Average Queue Size	100.1961	99.5810	102.1215
STD Queue	7.2533	8.3069	18.0892
Loss Rate	0.0797	0.0830	0.0732
Throughput	286.6679	286.6779	286.6879

4.2 Simulation 2

In this simulation we evaluate the robustness of the INN controller against variations in the network parameters. The numbers of sessions randomly change

by making some of them on/off during the simulations. Also, the sending rate is changed during the simulations. The number of TCP flows N is considered as a normally distributed random signal with mean 50, standard variance 10. Moreover, the link capacity is considered as a pulse signal, in this case it is increased from 2.4 Mb/s to 3Mb/s at 25th second and is decreased to 2.4 Mb/s at 50th second. Once more it is decreased to 1.8 Mb/s at 75th second. Queue regulation and drop probability, window size and RTT are shown in Fig. 4. As it is shown, good queue regulation of the PI is in the expense of misusing the resources. The PI controller and RED algorithm have constant oscillatory behavior. The INN controller regulates the queue response without overshoot and the least variance.

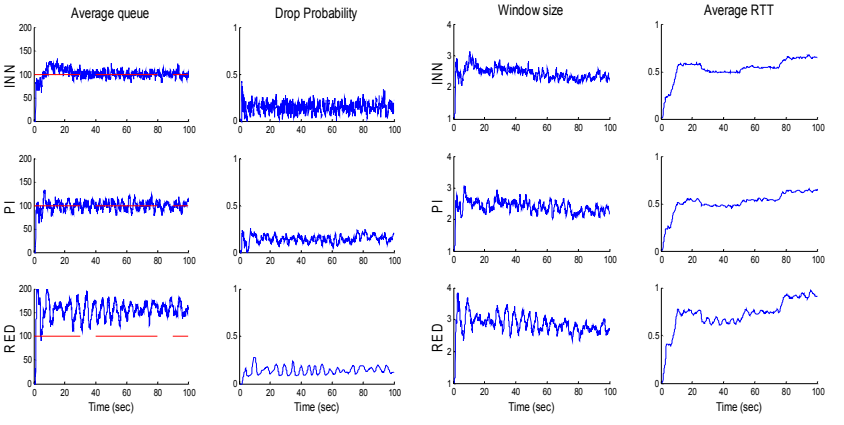


Fig. 4 Comparison of three AQM algorithms in time-variant condition

According to the formula (2), variations in link capacity causes step variations on queue lengths. Consequently, variations on queue size and RTT make Jitter in TCP network. Thus, our throughput in the network decayed. As it is shown in RED, the queue size is regulated in higher value than desired one and it has higher overshoot than the PI and INN controllers. And it is caused a smaller drop probability as compared with that of both controllers. Although, queue size regulation in the PI is done appropriately, the drop probability has higher value. The INN controller preserves the better performance rather than the PI and RED. In Table 2, the average queue size, STD queue, loss rate and throughput are shown.

Table 2 Comparison of three AQM algorithms in time-variant condition

	INN	PI	RED
Average Queue Size	100.2455	99.3711	129.6914
STD Queue	5.9310	11.8798	12.8271
Loss Rate	0.1399	0.1450	0.1155
Throughput	287.1779	287.1679	287.1779

4.3 Simulation 3

In this simulation, in addition to change the number of users and link capacity, we consider that %5 of output packets of router is lost by noise. The used TCP Reno algorithm could not diagnosis more than one error for each RTT; consequently, throughput in network is decreased. In Fig. 5 queue regulation using the INN controller is significantly better than the PI and RED. Simulation results are depicted in Fig. 5.

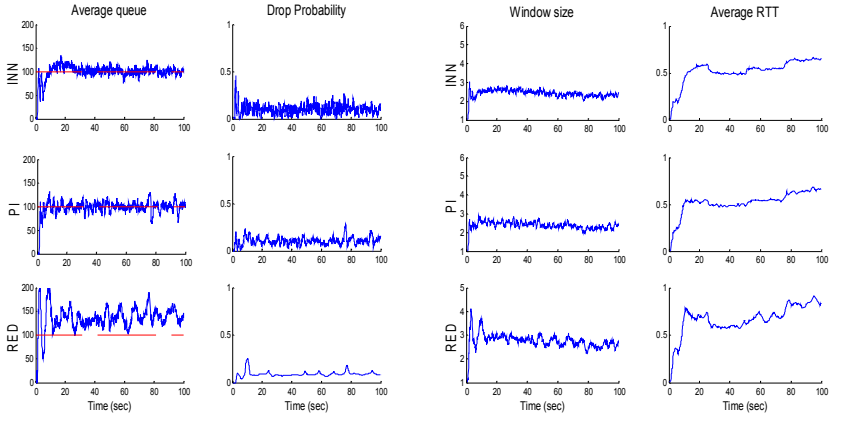


Fig. 5 Comparison of three AQM algorithms in addition noise to system

The queue length variation in the PI controller is much higher than INN controller. In the case of RED AQM, the queue length mean is far from the desired value. Consequently, the drop probability is smaller than both controllers. In Table 3 the average queue size, STD queue, loss rate and throughput are shown.

Table 3 Comparison of three AQM algorithms in addition noise to system

	INN	PI	RED
Average Queue Size	100.7066	97.9920	122.1140
STD Queue	6.3099	10.2066	15.8948
Loss Rate	0.0897	0.0953	0.0704
Throughput	286.4679	286.4779	286.7379

4.4 Simulation 4

In this simulation, in addition to change the number of users, link capacity and noise, we consider effect of unresponsive flows on performance of three used AQM schemes. UDP flows do not respond to the data lost because it does not have any acknowledgment. These flows decrease the network bandwidth, thus, the throughput decreases. Simulation results are depicted in Fig. 6.

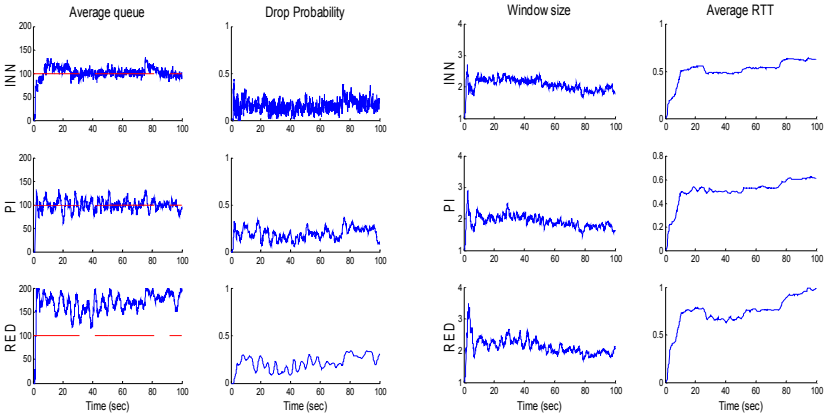


Fig. 6 Comparison of three AQM algorithms in addition unresponsive flows to system

As it is shown in Fig. 6 the predictive INN controller preserves a better performance than the PI and RED. Queue length with the PI controller is very fluctuant and it is mutable and far from the desired value using RED controller. Table 4 shows the average queue size, STD queue, loss rate and throughput.

Table 4 Comparison of three AQM algorithms in addition unresponsive flows to system

	INN	PI	RED
Average Queue Size	101.3544	99.4575	169.8199
STD Queue	7.6790	6.8038	17.4456
Loss Rate	0.1342	0.1809	0.1382
Throughput	291.1382	291.7183	291.9883

5 Conclusions

The neural network controller can promptly adapt its operation to the nonlinear time-varying and stochastic nature of TCP networks. We derived a learning procedure by adding *Delta-Bar-Delta* algorithm to the gradient descent BP method.

We applied a normal range of disturbances such as noise and unresponsive flows, and usual uncertainty such as variable user numbers, variable link capacity, to the three AQM methods: RED, PI and INN and compared their operations on congestion conditions. We showed by adding the disturbance to TCP network plant, some of the factors in network change, and it is a good reason for congestion in TCP networks.

Then we showed that INN as a heuristic and adaptive controller is able to control the revolted system. Moreover, our simulations show that the performance of the INN controller in desired queue tracking as well as disturbance and uncertainty rejection is high. In addition, the INN controller is better than classic

methods when there are a normal range of disturbance and uncertainty in the system.

In the future work we will analyze the effect of higher disturbances and sudden uncertainty on the proposed AQM method. We also try to exploit the benefits of adaptive and classic controllers in a combined method.

Acknowledgments

The authors sincerely thank the Iran Telecommunication Research Center (ITRC) for their financial support under grant number 11547.

References

[1] Hosseini, H., Shabanian, M., Araabi, B.: A Neuro-Fuzzy Control for TCP Network Congestion. In: WSC 2008 Conference (2008)
[2] Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ACM Trans. on Networking 1, 397–413 (1993)
[3] Hollot, C.V., Misra, V., Towsley, D., Gong, W.B.: A Control Theoretic Analysis of RED. In: Proc. of IEEE INFOCOM, pp. 1510–1519 (2001)
[4] Hollot, C.V., Misra, V., Towsley, D., Gong, W.B.: Analysis and design of controllers for AQM routers supporting TCP flows. IEEE Trans. on Automatic Control 47, 945–959 (2002)
[5] Sun, C., Ko, K.T., Chen, G., Chen, S., Zukerman, M.: PD-RED: To improve the performance of RED. IEEE Communication Letters 7, 406–408 (2003)
[6] Ryu, S., Rump, C., Qiao, C.: A Predictive and robust active queue management for Internet congestion control. In: Proc. of ISCC 2003, pp. 1530–1346 (2003)
[7] Zhang, H., Hollot, C.V., Towsley, D., Misra, V.: A self-tuning structure for adaptation in TCP/AQM networks. In: Proc. of IEEE/GLOBECOM 2003, vol. 7, pp. 3641–3646 (2003)
[8] Hadjadj, Y., Nafaa, A., Negru, D., Mehaoua, A.: FAFC: Fast Adaptive Fuzzy AQM Controller for TCP/IP Networks. IEEE Trans. on Global Telecommunications Conference 3, 1319–1323 (2004)
[9] Taghavi, S., Yaghmaee, M.H.: Fuzzy Green: A Modified TCP Equation-Based Active Queue Management Using Fuzzy Logic Approach. In: Proc. of IJCSNS, vol. 6, pp. 50–58 (2006)
[10] Hadjadj, Y., Mehaoua, A., Skianis, C.: A fuzzy logic-based AQM for real-time traffic over internet. Proc. Computer Networks 51, 4617–4633 (2007)
[11] Cho, H.C., Fadali, M.S., Lee, H.: Neural Network Control for TCP Network Congestion. In: Proc. American Control Conference, vol. 5, pp. 3480–3485 (2005)
[12] Jang, J.R., Sun, C., Mizutani, E.: Neuro-Fuzzy and Soft Computing. Prentice-Hall, Englewood Cliffs (1997)
[13] Quet, P.F., Ozbay, H.: On the design of AQM supporting TCP flows using robust control theory. IEEE Trans. on Automatic Control 49, 1031–1036 (2004)
[14] Misra, V., Gong, W.B., Towsley, D.: Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In: Proc. of ACM/SIGCOMM, pp. 151–160 (2000)

RAM Analysis of the Press Unit in a Paper Plant Using Genetic Algorithm and Lambda-Tau Methodology

Komal*, S.P. Sharma, and Dinesh Kumar

Abstract. Reliability, availability and maintainability (RAM) analysis gives some idea to carryout design modifications, if any, required to achieve high performance of the complex industrial systems. In the present study two important tools namely Lambda-Tau methodology and genetic algorithms are used to build the hybridized technique GABLT (Genetic Algorithms based Lambda-Tau) for RAM analysis of these systems. Expressions of reliability, availability and maintainability for the system are obtained by using Lambda-Tau methodology and genetic algorithm is used to construct the membership function. A general RAM index is used for further analysis. Fault tree is used to model the system. The proposed approach has been applied to the press unit of the paper plant situated in north India producing 200 tons of paper per day. The computed results are presented to plant personnel for their active consideration. The results will be very helpful for plant personnel for analyzing the system behavior and to improve the system performance by adopting suitable maintenance strategies.

Keywords: Reliability; Availability; Maintainability; Lambda-Tau methodology; Genetic algorithms.

1 Introduction

In a production plant, to obtain maximum output it is necessary that each of its subsystem/unit should run failure free and furnish excellent performance to achieve desired goals [1]. High performance of these units can be achieved with highly reliable subunits and perfect maintenance. Perfect maintenance means large capital input in the plant. So management personnel want minimum perfect maintenance to achieved

Komal · S.P. Sharma
Department of Mathematics, Indian Institute of Technology Roorkee(IITR), Roorkee, Uttarakhand, 247667, India

Dinesh Kumar
Department of Mechanical and Industrial Engineering, Indian Institute of Technology Roorkee(IITR), Roorkee, Uttarakhand, 247667, India
e-mail: karyadma.iitr@gmail.com

* Corresponding author.

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 127–137.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

desired goals at minimum cost. So maintainability aspects must be included to achieve customer satisfaction and remain competitive. To this effect the knowledge of behavior of system, their component(s) is customary in order to plan and adapt suitable maintenance strategies. The behavior of such systems can be studied in terms of their reliability, availability and maintainability (RAM)[2]. RAM as an engineering tool evaluates the equipment performance at different stages in design process. The information obtained from analysis helps the management in assessment of the RAM needs of system. Factors that affect RAM of a repairable industrial system include machinery operating conditions, maintenance conditions, and infra-structural facilities [3]. Analytical models become too complex for analyzing the interplay of the many different factors affecting RAM of repairable industrial systems. As industrial systems become more complex, it is not so easy to calculate RAM using collected or available data for the system as this is imprecise and vague due to various practical reasons. As these systems are repairable, failure rate and repair time of their components are used to estimate RAM of these system. But it is a common knowledge that large quantity of data is required in order to estimate more accurately, the failure and repair rates. However, it is usually impossible to obtain such a large quantity of data in any particular plant due to rare event of component's failure, human errors and economic restraints. These challenges imply that a new and pragmatic approach is needed to access and analyze RAM of these systems because organizational performance and survivability depends a lot on reliability and maintainability of its components/parts and systems. Knezevic and Odoom [4] gave some idea to compute reliability and availability of the system by using limited, imprecise and vague data. But the problem through this approach is that as the number of components of the system increases or system structure becomes more complex, the calculated reliability indices in the form of fuzzy membership function have wide spread i.e high uncertainty due to various arithmetic operations used in the calculations [5,6]. It means these indices have higher range of uncertainty and can't give exact idea about the system's behavior. To reduce the uncertainty level, spread for each reliability index must be reduced so that plant personnel can use these indices to analyze the system's behavior and take more sound decisions to improve the performance of the plant. This suggests that spread of each reliability index must be optimized. Mon and Cheng [7] suggested a way to optimize the spread of fuzzy membership function using available software package. Variety of methods also exists for optimization and applied in various technological fields for various purposes [8-10]. Genetic algorithms (GA) have been widely used to solve the optimization problems [11,12]. Arslan and Kaya[13] used genetic algorithm to determine the membership functions in a fuzzy system. Huang et. al.[14] used GA for Bayesian reliability membership function construction using fuzzy life time data. GA methods, which are basically random search techniques have been applied to many different problems like function optimization, routing problem, scheduling, design of neural networks, system identification, digital signal processing, computer vision, control and machine learning. Thus to optimize the spread of each computed fuzzy membership function up to a desired accuracy, GA can be used.

Thus it is observed from the study that by using limited, vague and imprecise data of the system, RAM parameters may be calculated. The objective of the present investigation is to develop an approach for assessing the effect of failure

pattern on a composite measure of RAM of a repairable system. In this paper, RAM analysis of the press unit of a paper industry is carried out using GA and Lambda-Tau methodology.

2 GABLT Technique

Assumptions of the model are:

- component failures and repair rates are statistically independent, constant, very small and obey exponential distribution function;
- the product of the failure rate and repair time is small (less than 0.1);
- after repairs, the repaired component is considered as good as new.

Here statistically independent means that the failure and repair of one component does not affect the failure and repair of other component but the overall performance of the system is affected [3]. Knezevic and Odoom [4] utilize the concept of Lambda-Tau methodology and coupled it with fuzzy set theory and Petri nets (PN). PN methodology is used to model the system while fuzzy set theory takes care of impreciseness in the data. Table 1 gives the basic expressions for failure rate and repair time associated with the logical AND-gates and OR-gates [15] used by them. Table 2 gives reliability indices for the repairable system RAM analysis used in the present study [3,16].

Table 1 Basic expressions of Lambda –Tau methodology

Gate	λ_{OR}	τ_{OR}	λ_{AND}	τ_{AND}
Expressions for n-inputs	$\sum_{i=1}^{n} \lambda_i$	$\frac{\sum_{i=1}^{n} \lambda_i \tau_i}{\sum_{i=1}^{n} \lambda_i}$	$\prod_{j=1}^{n} \lambda_j \left[\sum_{i=1}^{n} \prod_{\substack{j=1 \\ i \neq j}}^{n} \tau_j \right]$	$\frac{\prod_{i=1}^{n} \tau_i}{\sum_{j=1}^{n} \left[\prod_{\substack{i=1 \\ i \neq j}}^{n} \tau_i \right]}$

Table 2 Reliability indices used for RAM analysis

Reliability indices	Expressions
Reliability	$R_s = e^{-\lambda_s t}$
Availability	$A_s = \frac{\mu_s}{\lambda_s + \mu_s} + \frac{\lambda_s}{\lambda_s + \mu_s} e^{-(\lambda_s + \mu_s)t}$
Maintainability	$M_s = 1 - e^{(-\mu_s t)}$

But the problem through this approach is that as the number of components in the system increases or system structure become more complex, the calculated reliability indices in the form of fuzzy membership function have wide spread due to various arithmetic operations involved in the calculations [5]. GABLT overcome these problems and strategy followed through this approach is shown in Fig. 1. In GABLT technique system is modeled with the help of fault tree by finding min-cut of the system and expressions for RAM are calculated using Lambda-Tau methodology. Since data for repairable industrial systems in the form of λ and τ are uncertain so taken as known fuzzy numbers (triangular). Fault-tree and fuzzy set theory can be seen in hybrid form in literature [17,18]. By using fuzzy λ and τ, upper boundary value of RAM indices are computed at cut level α in the process of membership function construction by solving the optimization problem(A).

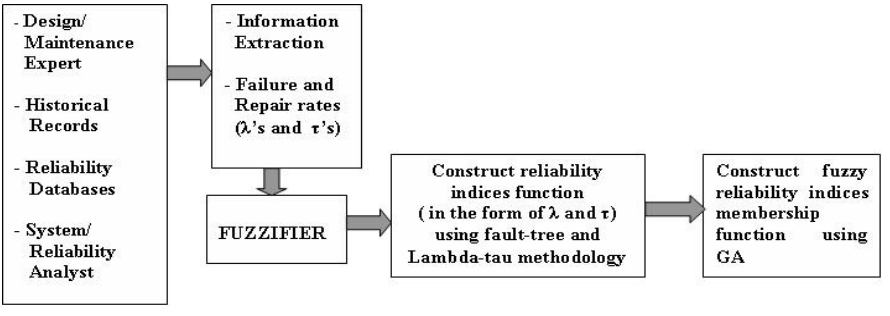


Fig. 1 Flow chart of GABLT technique

Problem:

Maximize/Minimize: $\tilde{F}\ (t\ /\ \lambda_{1,}, \lambda_2, ...,\ \lambda_n, \tau_1, \tau_2, ...,\ \tau_m)$

Subjected to: $\mu_{\lambda_i}(x) \geq \alpha$ .....(A)

$\mu_{\tau_j}(x) \geq \alpha$

$0 \leq \alpha \leq 1$

$i = 1,2,.....n, \qquad j = 1,2,....m$

The obtained maximum and minimum values of F are denoted by F_{max} and F_{min} respectively.

The membership function values of $\tilde{F}$ at F_{max} and F_{min} are both α that is,

$$\mu_{\tilde{F}}(F_{\max}) = \mu_{\tilde{F}}(F_{\min}) = \alpha$$

Where $\tilde{F}(t/\lambda_1, \lambda_2 ..., \lambda_n, \tau_1, \tau_2, ..., \tau_m)$ is the time dependent fuzzy reliability index. Since the problem is of non-linear in nature and needs some effective techniques and tools available in literature to solve.

In this paper GA [11] is used as a tool to find out the optimal solution of the above optimization problems. In the present analysis binary coded GA is used. The objective function for maximization problem and the reciprocal of the

objective function for minimization problem is taken as the fitness function. Roulette-wheel selection process is used for reproduction. One-point crossover and random point mutation are used in the present analysis. To stop the optimization process maximum number of generations and change in population fitness value are used.

3 RAM Index

Rajpal et al.[2] used a composite measure of reliability, availability and maintainability for measuring the system performance and named as RAM index. They used specific values of these parameters to calculate RAM index. In this study a time dependent ram index has been proposed and given in equation (1).

$$RAM=W_1\times R_s(t) + W_2\times A_s(t) + W_3\times M_s(t) \quad (1)$$

Where $W_i \in (0,1)$,i=1,2,3. are weights such that $\sum_{i=1}^{3} W_i = 1$. Rajpal et al.[2] used W=[0.36,0.30,0.34] to calculate the RAM index. Same values of W's are used here. In this study reliability, availability and maintainability are in the form of fuzzy membership functions so RAM index itself comes as a fuzzy membership function in the form of triplet, $\widetilde{RAM}$ =(RAM_L,RAM_M,RAM_R). The crisp value of the fuzzy membership function is given in equation (2).

$$RAM=RAM_L+[(RAM_M-RAM_L)+(RAM_R-RAM_L)]/3 \quad (2)$$

This suggests that RAM$\in$ (0,1).

4 An Illustration with Application

Kumar [1] have analyzed and optimized system availability in sugar, paper and fertilizer industries. In the present study a paper plant situated in north India producing 200 tons of paper per day, is considered as the subject of discussion. The paper plants are large capital oriented engineering system, comprises of subsystems namely chipping, feeding, pulping, washing, screening, bleaching, production of paper consisting of press unit and collection, arranged in complex configuration. The actual papermaking process consists of two primary processes: dry end operations and wet end operations. In wet end operations, the cleaned and bleached pulp is formed into wet paper sheets. In the dry end operations, those wet sheets are dried and various surface treatments are applied to the paper. The main components of a paper machine are headbox, wire section, press unit and drying section. Press unit, an important functionary part of the paper plant which has a dominant role in production of the paper is taken as the main system. Press unit consists of felt and top and bottom rolls as its main components. The unit receives wet paper sheet from the forming unit on to the felt, which is then carried through press rolls and thereby reducing the moisture content to the extent.

The system consists of seven subunits with their fault tree shown in Fig. 2, where PSF, press system top failure event; E1, subsystem 1, felt; E2, subsystem 2, top roller;E3, subsystem 3, bottom roller; i = 1, felt; i = 2, 5, top, bottom roller bearing; i = 3, 6, top, bottom roller bending; i = 4, 7, top, bottom roller rubber wear. Under the information extraction phase, the data related to failure rate (λ_i) and repair time (τ_i) of the components is collected from present/historical records of the paper mill and is integrated with expertise of maintenance personnel as presented in Table 3 [1, 19].

Table 3 Press unit component's failure rate and repair time

Press unit
$\lambda_1 = 1\times 10^{-4},\ \lambda_2 = \lambda_5 = 1\times 10^{-3},\ \lambda_3 = \lambda_6 = 1.5\times 10^{-3},$
$\lambda_4 = \lambda_7 = 2\times 10^{-3}$ (in failure /hrs)
$\tau_1 = 5,\ \tau_2 = \tau_5 = 2,\ \tau_3 = \tau_6 = 3,\ \tau_4 = \tau_7 = 4$ (in hrs)

The data are imprecise and vague. To account the imprecision and uncertainty in the data, crisp input is converted into known triangular fuzzy number with ±15% spread. As a case the failure rate and repair time for the felt (A) is shown in Fig. 3.

GABLT technique has been applied and values of selected parameters for GA are given as:

Population size: 140
Chromosome size: 40
Probability of crossover (P_c): 0.8
Probability of mutation (P_m): 0.005
Number of generations: 50

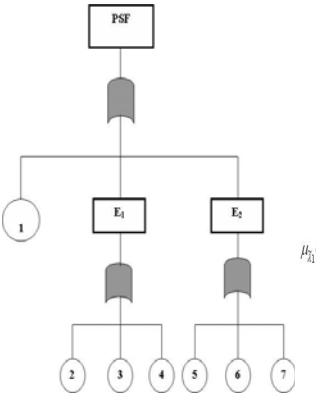


Fig. 2 Press unit fault tree

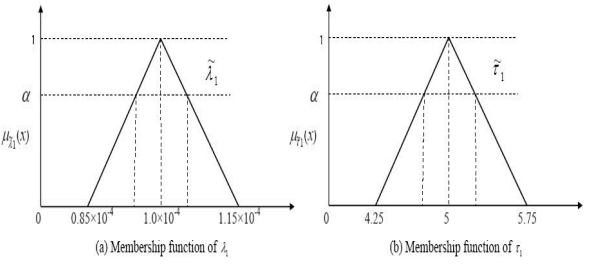


Fig. 3 Input data for GABLT corresponding to felt

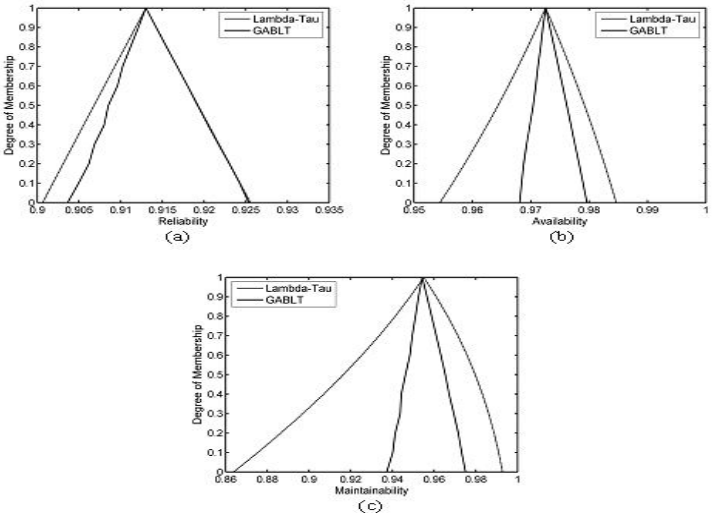


Fig. 4 RAM of the press unit at time t=10 hrs

For mission time 10 (hrs), computed reliability, availability and maintainability of the system have been plotted and shown in Fig. 4 along with Lambda-Tau results. The result shows that GABLT results have smaller spread i.e. lesser uncertainty and range of prediction.

This suggests that GABLT is a better approach than Lambda-Tau. At different α-cuts (0,0.5,1) system's reliability, availability and maintainability curve for 0–50(hrs) have been plotted by using GABLT technique and shown in Fig. 5 along with their membership function at t=40, 15 and 10(hrs) respectively to show the behavior with different level of uncertainties. To see the behavior of RAM index against different uncertainty (spread) levels, a plot between spread from 0 to 100(in %) and RAM index has been plotted and shown in Fig. 6. Figure shows that as uncertainty level increases RAM index decreases i.e. to achieve higher performance of the system uncertainties should be minimize. Performance of the system directly depends on each constituent subunits/components. So to check the performance and to analyze the effect of variation in failure rate and repair time of four main components of the press unit i.e. felt, roller bearing, roller bending and roller rubber wear at t=10 hrs by varying their failure rate and repair time each at a time and fixing failure rate and repair time of other components at the same time, on RAM index of the system are computed and shown in Fig. 7. Figure contains four subplots corresponding to four main components of the press unit. Each subplot contains two subplots against variation in failure rate and repair time respectively of the corresponding component without increase in other component's failure rate and repair time. These plots show that when failure rate and repair time increases for felt then variation in RAM index of the system is almost same but for other components, RAM index decrease rapidly and corresponding maximum and minimum values are given in Table 4.

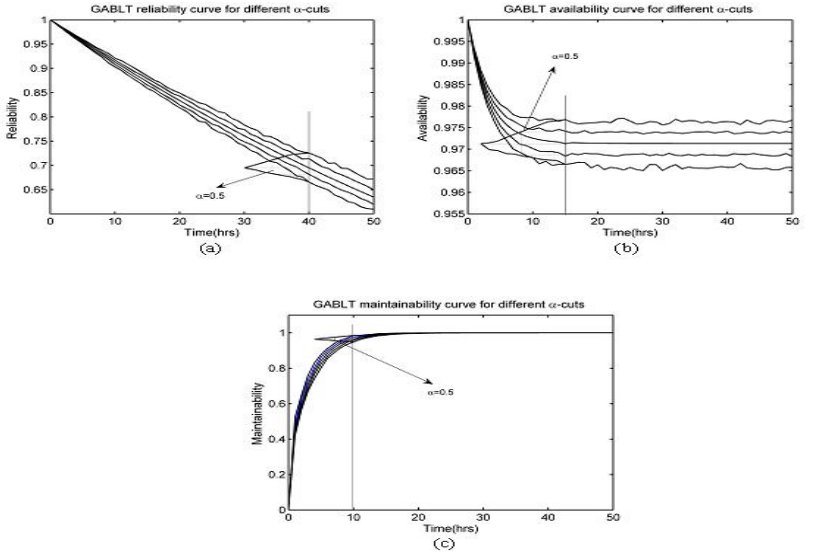


Fig. 5 Behavior of RAM of the press unit for long run time period at different level of uncertainties

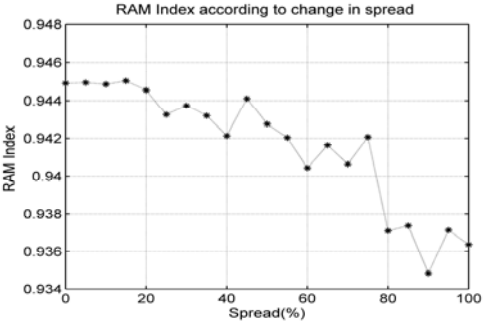


Fig. 6 Variation in RAM index by varying uncertainty (spread) level from 0 to 100 (%)

5 Discussion and Conclusion

Fig. 4 clearly shows that GABLT technique results have smaller spread than Lambda-Tau results because GA provides solution near to optimal solution. The behavior of reliability, availability and maintainability of the system for long run period (0-50 hrs) using current conditions and uncertainties are shown by plotting these parameters in Fig. 5. Result shows that if current condition of equipments and subsystems are not changed then reliability of the system decreased rapidly while availability and maintainability behaves almost linear after certain time for long run period. Fig.7 and Table 4 suggest that to optimize RAM index of the system, failure rate and repair time of its constituent components should be decrease.

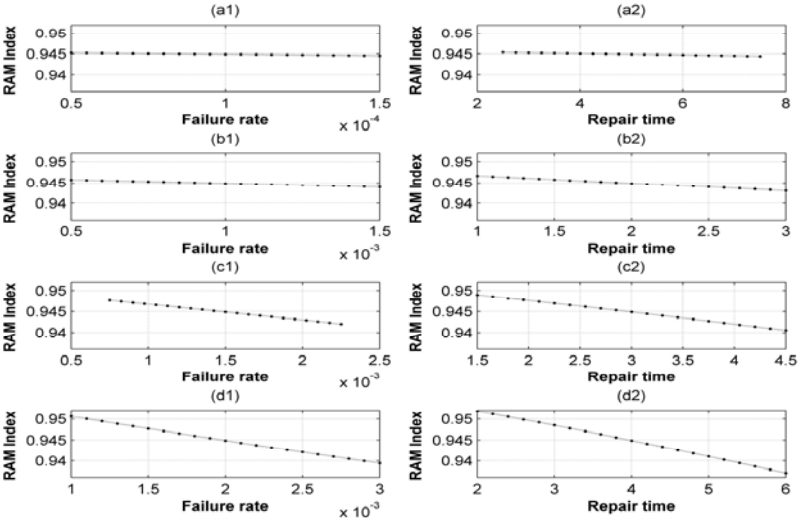


Fig. 7 Effect on RAM index by varying failure rate and repair time corresponding to four main components of press Unit (a) Felt (b) Roller bearing (c) Roller bending (d) Roller rubber wear

Table 4 Effect of variations in failure rate and repair time of components on RAM index

Component	Range of failure rate ($\times 10^{-3}$)	RAM index	Range of repair time	RAM index
Felt	0.05-0.15	Max:0.94527 Min: 0.94453	2.5-7.5	Max:0.94536 Min: 0.94443
Roller bearing	0.50-1.5	0.94576080 0.94390878	1.0-3.0	0.946728113 0.943010087
Roller bending	0.75- 2.25	0.94766977 0.94209396	1.5-4.5	0.94893066 0.940582087
Roller rubber wear	1.0 – 3.0	0.95062793 0.93944597	2.0-6.0	0.951832492 0.937071309

As an example if failure rate and repair time of roller rubber wear are deceased up to 50% then RAM index means performance of the system increased up to 0.8% of the current value. On the basis of results tabulated, it is analysed that to improve the performance of the press unit, more attention should be given to the components in order roller rubber wear, roller bending, roller bearing and felt. These results of press unit will help the concern managers to plan and adapt suitable maintenance practices/strategies for improving system performance and thereby reduce operational and maintenance costs. Thus, it will facilitate the management in reallocating the resources, making maintenance decisions, achieving long run availability of the system, and enhancing the overall productivity of the paper industry. Apart from these advantages, the system performance analysis

may be utilized to conduct cost-benefit analysis, operational capability studies, inventory/spare parts management, and replacement decisions.

6 Managerial Implications Drawn

This paper reports the reliability, availability and maintainability of the press unit in a paper mill by using GABLT technique a better approach than Lambda-Tau. This technique optimizes the spread of reliability indices indicating higher sensitivity zone and thus may be useful for the reliability engineers/experts to make more sound decisions. Plant personnel's may be able to predict the system behavior more precisely and will plan future maintenance.

In nutshell the important managerial implications drawn by using the discussed technique are:

- to model and predict the behavior of industrial systems in more consistent manner;
- to improve the performance of the press unit, more attention should be given to the components in order roller rubber wear, roller bending, roller bearing and felt ;
- to determine reliability characteristics (such as MTBF,MTTR) important for planning the maintenance need of the systems;
- to plan suitable maintenance strategies to improve system performance and reduce operation and maintenance costs.

Acknowledgement

The authors are thankful to the anonymous referees for their valuable comments and suggestions. Also, the corresponding author (Komal) acknowledges the Council of scientific and industrial research (CSIR), New Delhi, India for all financial support to carry out the research work.

References

[1] Kumar, D.: Analysis and optimization of systems availability in sugar, paper and fertilizer industries, PhD thesis, Roorkee(India), University of Roorkee (1991)

[2] Rajpal, P.S., Shishodia, K.S., Sekhon, G.S.: An artificial neural network for modeling reliability, availability and maintainability of a repairable system. Reliability Engineering & System Safety 91, 809–819 (2006)

[3] Ebling, C.E.: An introduction to reliability and maintainability engineering. Tata McGraw Hill Publishing Company Limited, New Delhi (1997)

[4] Knezevic, J., Odoom, E.R.: Reliability Modeling of repairable system using Petri nets and fuzzy Lambda-Tau methodology. Reliability Engineering & System Safety 73, 1–17 (2001)

[5] Chen, S.M.: Fuzzy system reliability analysis using fuzzy number arithmetic operations. Fuzzy Sets and Systems 64(1), 31–38 (1994)

[6] Komal, Sharma, S.P., Kumar, D.: Reliability analysis of the feeding system in a paper industry using lambda-tau technique. In: Proceedings of International Conference on Reliability and Safety Engineering (INCRESE), India, pp. 531–537 (2007)
[7] Mon, D.L., Cheng, C.H.: Fuzzy system reliability analysis for components with different membership functions. Fuzzy Sets and Systems 64(1), 145–157 (1994)
[8] Tillman, F.A., Hwang, C.L., Kuo, W.: Optimization of System Reliability. Marcel Dekker, New York (1980)
[9] Ravi, V., Murty, B.S.N., Reddy, P.J.: Non equilibrium simulated annealing-algorithm applied to reliability optimization of complex systems. IEEE Transactions on Reliability 46, 233–239 (1997)
[10] Ravi, V., Reddy, P.J., Zimmermann, H.J.: Fuzzy Global Optimization of Complex System Reliability. IEEE Transactions on Fuzzy systems 8(3), 241–248 (2000)
[11] Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading (1989)
[12] Coit, D.W., Smith, A.: Reliability optimization of series-parallel systems using a genetic algorithm. IEEE Transactions on Reliability 45(2), 254–260 (1996)
[13] Arslan, A., Kaya, M.: Determination of fuzzy logic membership functions using genetic algorithms. Fuzzy Sets and Systems 118, 297–306 (2001)
[14] Huang, H.Z., Zuo, M.J., Sun, Z.Q.: Bayesian reliability analysis for fuzzy lifetime data. Fuzzy Sets and Systems 157, 1674–1686 (2006)
[15] Singh, C., Dhillion, B.S.: Engineering Reliability: New Techniques and Applications. Wiley, New York (1991)
[16] Tanaka, H., Fan, L.T., Lai, F.S., Toguchi, K.: Fault-tree analysis by fuzzy probability. IEEE Transactions Reliability 32, 453–457 (1983)
[17] Singer, D.: A fuzzy set approach to fault tree and reliability analysis. Fuzzy Sets and Systems 34, 45–155 (1990)
[18] Sawyer, J.P., Rao, S.S.: Fault tree analysis of fuzzy mechanical systems. Microelectron and Reliability 34(4), 653–667 (1994)
[19] Sharma, R.K., Kumar, D., Kumar, P.: Modeling System Behavior for Risk and Reliability Analysis using KBARM. Quality and Reliability Engineering International 23(8), 973–998 (2007)

A Novel Approach to Reduce High-Dimensional Search Spaces for the Molecular Docking Problem

Dimitri Kuhn, Robert Günther, and Karsten Weicker

Abstract. Molecular simulation docking has become an important contribution to pharmaceutical research. However, in the case of fast screening of many substances (ligands) for their potential impact on a pathogenic protein, computation time is a serious issue. This paper presents a technique to reduce the search space by keeping the ligands close to the surface of the protein.

1 Introduction

The development of novel drugs is a challenging and costly process in pharmaceutical research. One key step of this process is the screening of large libraries of chemical compounds for potential drug candidates interacting with a particular protein or pathogen, often referred as target. A viable alternative to the costly experimental screening by robots are virtual screening (VS) techniques employing computational molecular docking methods given the 3-D structure of the protein target is known. The major advantage of these methods is to provide detailed information on the interactions between the protein and the small compound (ligand). These information can subsequently be used to improve potential drug candidates by computer aided rational drug design.

Dimitri Kuhn
HTWK Leipzig University of Applied Science
e-mail: dkuhn@imn.htwk-leipzig.de

Robert Günther
University of Leipzig, Institute of Biochemistry, Brüderstraße 34, 04103 Leipzig, Germany
e-mail: robguent@uni-leipzig.de

Karsten Weicker
HTWK Leipzig University of Applied Science, FbIMN, Gustav-Freytag-Str. 42A, 04277 Leipzig, Germany
e-mail: weicker@imn.htwk-leipzig.de

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 139–148.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

To solve the molecular docking problem, several approaches have been developed [4]. Though computational expensive, molecular docking simulations have proven to be reliable techniques to predict a correct protein-ligand complex. Assuming that the native protein-inhibitor complex corresponds to that with the lowest calculated binding energy (E_{Bind}), the correct prediction of the resulting complex can be reformulated as an optimisation problem. In this context, a small molecule (ligand) is docked onto a protein (receptor) to find the optimal protein-ligand complex, which is characterised by the position, orientation and shape of the ligand (pose). While the receptor is represented as a rigid body, the ligand is treated as a flexible entity. More precisely, the binding energy between the ligand and the protein can be computed for each possible pose of the ligand and, thus, serves as an objective function for the minimisation process. However, the problem is to find the global minimum of the binding energy function, which corresponds to the optimal protein-ligand complex.

This problem was often tackled by heuristics, genetic algorithms [5], and other nature-inspired optimisation algorithms like particle swarm optimisations [6]. Though, quite successful in predicting the correct pose, several thousand evaluation steps are needed to find the solution. However, computation of E_{Bind} at each step is computational demanding. As a consequence, any new concept is welcome to tailor the algorithm to the problem in the sense of the no free lunch theorems [11].

Here, we introduce a novel approach to lower the computational cost of molecular docking simulation methods. In an preprocessing step the search space for an evolutionary algorithms is trimmed down. This technique reduces the number of unnecessary computing steps in the evolutionary optimization and leads to rapid convergence nearby the optimal solution.

2 Motivation

The bottleneck of molecular docking simulations methods is the evaluation of the binding energy E_{Bind} at each simulation step. In AutoDock3 [5] (AD3), which is one of the widely used docking programs of this type, E_{Bind} is computed according to eqn 1, where $_{i,j}$ is the distance of two atoms.

$$
\begin{aligned}
Bind = \quad & vdW \sum_{i,j} \frac{ij}{12} - \frac{ij}{6} \quad + \quad hbond \sum_{i,j} ( ) \quad \frac{ij}{12} - \frac{ij}{10} \\
+ \quad & elec \sum_{i,j} \frac{i \ j}{(\ ij) \ ij} \\
+ \quad & tor \quad tor + \quad sol \sum_{i,j} (\ i \ j + \ j \ i) \ - \frac{r_{ij}^2}{2\sigma^2}
\end{aligned}
\tag{1}
$$

The constants E_* denote empirically determined coefficients obtained by linear regression over protein-ligand complexes with known binding energies [5]. In detail, the three terms E_{vdW}, E_{hbond} and E_{elec} represent the *in vacuo* contributions of the binding energy: a 12-6-Lennard Jones dispersion/repulsion term; a directional 12-10 hydrogen bonding term, where $(\ )$ stands for a directional weight based on the angle between the probe atom and the target atom; and an electrostatic potential between the partial charges $_*$ of the ligand and the protein. As a measure for entropic factors, a term proportional to the number of rotatable bonds in the flexible ligand ($_{tor}$) is added. Finally, a desolvation term describing the energy needed to strip off water molecules upon binding is added based on Stouten parameters [10]. The sums in this binding energy equation run over the indices and , which are the atoms of both binding partners. For a protein, the number of atoms can count up to several thousand.

The binding energy can be divided into two parts: (a) the interaction energy between the protein and the ligand (intermolecular energy) and (b) the internal energy of the small molecule (intramolecular energy). To save computing steps, the intermolecular energy can be pre-calculated by a grid based approach (see below). The actual intermolecular binding energy of the ligand can then be determined by fast trilinear interpolation. Details on the constants in eqn 1 and the description of the computing process can be found in [5].

Figure 1 shows a standard problem instance – the HIV-1 protease. Here it can be seen that the interesting places for the ligand cover only a small fraction of the search space – a small corridor atop of the molecule's

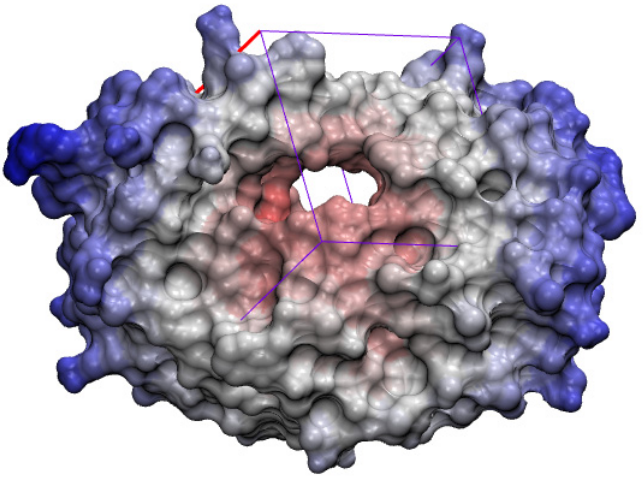


Fig. 1 The protein HIV-1 protease (pdb entry 1hvr, [2]) in surface representation. The binding site is enclosed in a box representing the pre-computed grid maps

surface. However the optimisation algorithm has only little guidance to stay at the surface. The detailed analysis of the approach of [5] reveals, that the ligand spends 97.5% of the time inside the protein wasting computing steps.

The position of the ligand is a translation vector which places the ligand in the three-dimensional space of the protein and its orientation is represented as a quaternion. Thus, depending on the number of rotatable bonds in the ligand ($_{tor}$), the dimension of the search space is $3+4+$ $_{tor}$. Consequently, for docking of highly flexible ligands like peptides a high number of evaluation steps is needed to find the correct protein-ligand complex. However, in the context of VS studies, it is often sufficient to find a pose nearby the optimal solution (*hot spots*), which can then be refined later on by a different method.

If we turn our intention to a rather coarse search for such hot spots the original approach of AD3 appears to be inefficient. As a consequence, this paper investigates whether a reduction of the search space to the protein's surface might be a valuable improvement.

3 Search Space Reduction

3.1 Computing the Protein's Surface

Based on the pre-computed grid maps we can derive for the $\times$ $\times$ grid points whether they are inside or outside of the molecule. This technique was first developed by Kuhn [1] in this context. Using a modified breadth-first search, we can compute the surface as the set of faces between grid points inside and outside of the molecule. Each face is represented as the grid point inside the molecule and the normalised vector directing to the grid point outside the molecule.

The algorithm starts at a surface point and computes the adjacent faces on the surface for each point. The faces are than connected as a graph. Figure 2 shows that there are only three possibilities for each of the four directions. The corresponding algorithm is displayed in Figure 3 in pseudo code notation.

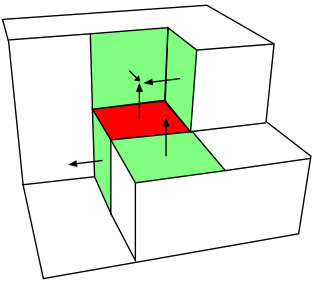


Fig. 2 Example for four adjacent sectors in the computation of the map

```
CREATE-MAP( rastered molecule G,  energy threshold b)
 1  map ← new  Container ()
 2  todo ← new  Queue ()
 3  r ←  find a protein point at the surface in  G
 4  r.neighbrs ← ∅
 5  todo.enqueue(r)
 6  while ¬todo.isEmpty()
 7  do ⌜ s ← todo.dequeue()
 8        ngh ← findneighbours(G, s, b)
 9        for each n ∈ ngh
10        do ⌜ if ¬map.contains(n)
11              then ⌜ if ¬todo.contains(n)
12                      then ⌜ s.neighbrs ← s.neighbrs ∪ {n}
13                             n.neighbrs ← n.neighbrs ∪ {s}
14                           ⌞ todo.enqueue(n)
15                      else ⌜ s.neighbrs ← s.neighbrs ∪ todo.find(n).neighbrs
16                             newneighbrs ← todo.find(n).neighbrs ∪ {s}
17           ⌞        ⌞      ⌞ todo.find(n).neighbrs ← newneighbrs
18     ⌞ map.insert(s)
19  return map
```

Fig. 3 Algorithm to compute the surface map

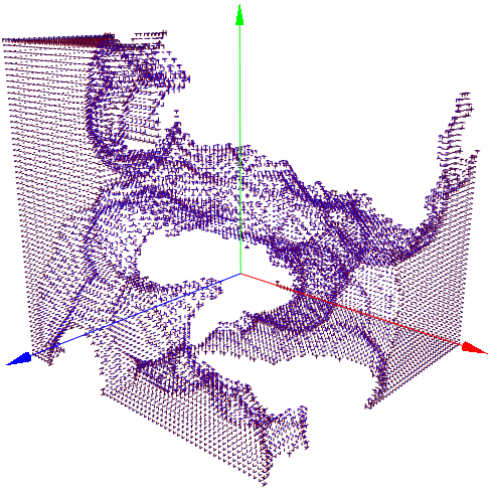


Fig. 4 Generated surface model of the HIV-1 protease

Our algorithm guarantees that each entry in the map belongs to the surface of the molecule. Note, that there might be various faces for the same grid point which necessarily differ in the normalised vector. The resulting surface for a cut-out of the HIV-1 protease protein is shown in figure 4. The direction of each normalised vector is indicated by a little arrow.

3.2 Optimising along the Surface

Concerning the optimisation phase, there are different feasible approaches.

- Application of a simple optimisation scheme to demonstrate the feasibility of the approach.
- Develop a well-adapted algorithm for fast optimisation within the surface graph.
- Construct an additional intermediate decoding step which enables the usage of a standard evolutionary algorithm on the graph representation.

Within the scope of this paper, we are interested in the general benefit of the approach for hot spot detection, i.e. we are interested in finding the regions of a protein with which the ligand might interact with a high probability. As a consequence we want a fast algorithm to place the ligand at promising positions – the detailed conformation of the ligand is of subordinate interest.

Therefore, we decided to follow a simple optimisation approach to show feasibility. We used a (30 180)-evolution strategy as the general course of the optimisation algorithm. The subsequent paragraphs present the key decisions concerning the genotype, the mutation operator, and the overall optimisation process.

Based on the description in the standard approach of AD3, each individual is represented by the face of the surface , a small offset to adjust the position of the ligand with respect to the face ( ), the quaternion for the orientation and the torsion angles $_i$ of the rotatable bonds. This is illustrated in figure 5.

Though we have increased the number of dimensions by 1 compared to the standard approach, we reduce the search space significantly by limiting it to a corridor around the surface. The mutation operator has to change all four elements of the genotype shown in figure 5. Concerning the offset, the quaternion, and the torsion angles we rely on the standard mechanism of

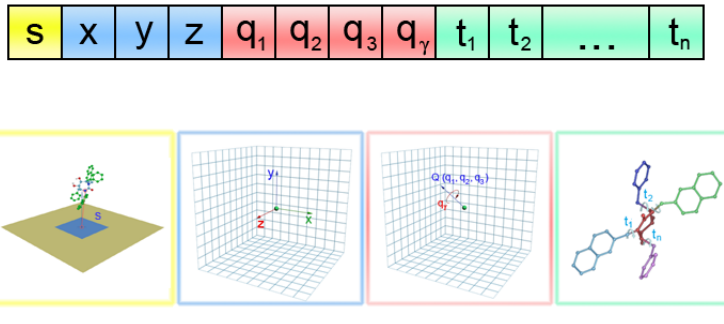


Fig. 5 Representation of the individual with the face s of the surface, an additional small offset (x, y, z), the quaternion q for the orientation of the ligand, and the torsion angles t_i

```
MUTATE-POSITION( individual A)
1  u ← generate random number from N(0, σ_face)
2  stepnumber ← ⌊|u|⌋
3  for i ← 1, ..., s
4  do ⌜ dir ← select uniform randomly direction
5     ⌞ A.position ← move (A.position, dir)
```

Fig. 6 Part of the mutation operator to change the position on the surface of the protein—the self-adaption of the strategy parameter σ_{face} is not shown

the evolution strategies [8, 7]. However, the face is changed according to the algorithm shown in figure 6, which is a random walk on the surface consisting of a randomly determined number of steps controlled by the parameter $_{facet}$.

In the first instance, we have experimented with fixed values for $_{face}$. But in most docking experiments better results could be reached with a self-adaptive step size $_{face}$ – additional to the step sizes for the other search space dimensions.

Note, that the effective step size is considerably smaller than the value in the algorithm since the random walk can reverse the direction. Nevertheless, the mechanism has proven to be a valid means to cover a broad area around the current face without anomalies in the distribution. A better integrated mutation operator for the position of the ligand is currently under development.

In a future intermediate optimisation step we will embed the surface graph in a two-dimensional plane. The aim is to assign each point of a two-dimensional grid to a node of the surface graph. Furthermore, there need to be at least one point for each node. The neighbouring structure should be reflected by the embedding function. Then we can use a two-dimensional standard evolution strategy to determine the point in the surface by interpolation. This enables standard step sizes instead of the rather unusual approach chosen by the current simple mutation approach.

4 Results

In this section we compare the results of our algorithm with the Lamarckian genetic algorithm by Morris et. al. [5] which uses the local optimiser of Solis and Wets [9].

The algorithms are tested on two standard benchmarks:

- Protein 1hvr (HIV-1 protease) with the ligand xk2A [2]. Here, the ligand has 10 torsion angles and the best known fitness value for E_{Bind} is -21.4 kcal/mol.
- Protein 3ptb (Beta-Trypsin) with ligand benA [3]. The ligand has 0 torsion angles and the best known fitness value is -8.18 kcal/mol.

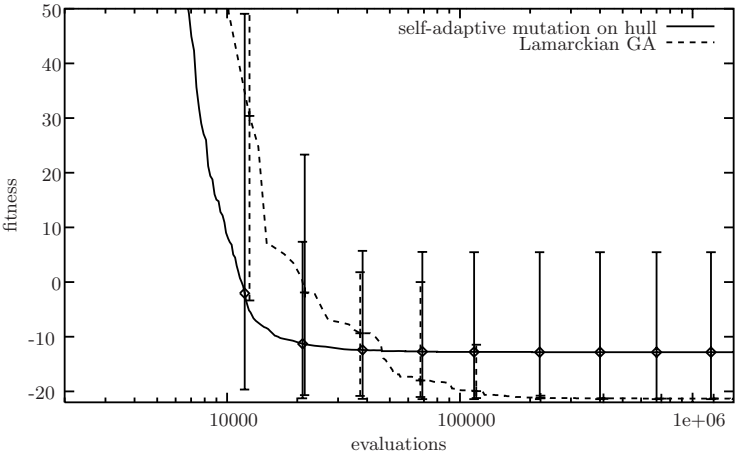


Fig. 7 Results for the 1hvr

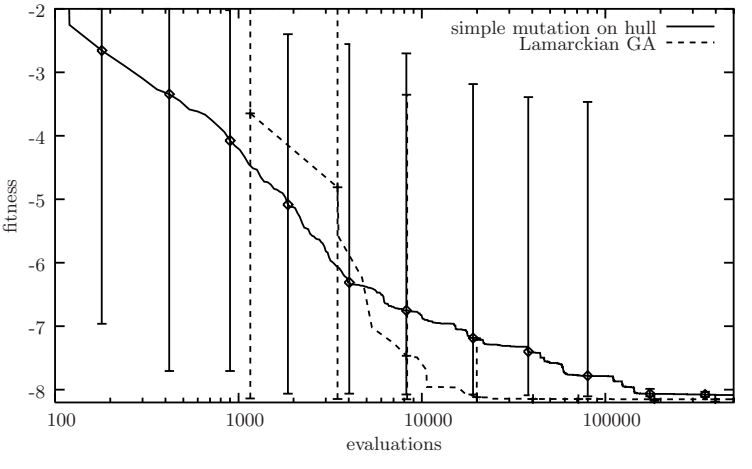


Fig. 8 Results for the 3ptb

For 1hvr the result of 50 optimisation runs using the Lamarckian GA as well as the self-adapting algorithm is shown in figure 7.

The self-adaptive algorithm reaches a fitness level of -10 kcal/mol already after 20,000 evaluations where the Lamarckian algorithm needs more than 35,000 evaluations. This relation holds also approximately for the fitness values for the best runs. This result shows the benefit of the new technique: approximately 50% time resources can be saved for hot spot detection. It also shows how well the adaptation of the torsion angles works together with the mutation of the current face.

The performance gain of the algorithm is less pronounced for ligands that only rely on the positioning of the ligand. The 3ptb problem is a good example

for this case. Figure 8 shows the results. Here we have chosen the simple mutation with a fixed value $_{face}$ which led to better results than the self-adaptive algorithm. On average, the algorithm is still slightly faster than the Lamarckian GA, but the fitness values of the best runs show clearly that the Lamarckian GA solves the problem better than the new algorithm using the surface of the protein. Since this problem instance relies on the positioning of the ligand only, this shows that the mutation operator needs to be improved. In fact, the approach to determine the actual maximum step size using a random variable $\mathcal{N}(0 \quad _{face})$ and then use this value for random steps leads to rather small step sizes.

5 Conclusion and Outlook

The general sensibility of the new approach has been shown by the experiments using the 1hvr protein. However, the results indicate also that the algorithm needs further improvement for two reasons:

- The positioning is not accurate enough to deal with a problem like 3ptb. As a consequence the surface graph is sought to be embedded in a two-dimensional plane which enables standard operators and an improved positioning of the ligand.
- An analysis of the invalid function evaluations shows that there are still 95.2% evaluations of ligands being partly inside of the protein. Therefore, a mechanism needs to be developed that keeps the ligands outside of the protein at first (using the normal vectors of the facets) and then starts to let the ligands sink slowly onto the protein's surface. Here are many alternative algorithms and approaches possible, which will be examined in the near future.

The main contribution of this article is the demonstration that the surface of a protein may be computed and used successfully to improve the docking process employing an adaptive evolutionary strategy. Considering the total number of evaluation steps needed to reach a promising pose nearby the optimal solution, our algorithm clearly performs better in both cases. Thus, if the focus is on pre-screen a huge library of up to several million compounds for promising candidates, the approach presented here can save computing time significantly. The resulting drug candidates can then be investigated employing other, even more computational demanding docking methods.

References

1. Kuhn, D.: Reduktion der Suchraumgröße bei der Simulation von Protein-Ligand-Wechselwirkungen am Beispiel von AutoDock. Master's thesis, HTWK Leipzig, Leipzig (2007)

2. Lam, P.Y., Jadhav, P.K., Eyermann, C.J., Hodge, C.N., Ru, Y., Bacheler, L.T., Meek, J.L., Otto, M.J., Rayner, M.M., Wong, Y.N.: Rational design of potent, bioavailable, nonpeptide cyclic ureas as hiv protease inhibitors. Science 263(5145), 380–384 (1994)
3. Marquart, M., Walter, J., Deisenhofer, J., Bode, W., Huber, R.: The geometry of the reactive site and of the peptide groups in trypsin, trypsinogen and its complexes with inhibitors. Acta Crystallographica Section B 39(4), 480–490 (1983), http://dx.doi.org/10.1107/S010876818300275X
4. Moitessier, N., Englebienne, P., Lee, D., Lawandi, J., Corbeil, C.R.: Towards the development of universal, fast and highly accurate docking/scoring methods: a long way to go. British Journal of Pharmacology 153(S1), S7–S26 (2008)
5. Morris, G.M., Goodsell, D.S., Halliday, R.S., Huey, R., Hart, W.E., Belew, R.K., Olson, A.J.: Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. Journal of Computational Chemistry 19, 1639–1662 (1998)
6. Namasivayam, V., Günther, R.: pso@autodock: A fast flexible molecular docking program based on swarm intelligence. Chemical Biology & Drug Design 70(6), 475–484 (2007)
7. Rechenberg, I.: Evolutionsstrategie 1994. Frommann-Holzboog, Stuttgart (1994)
8. Schwefel, H.P.: Evolution and Optimum Seeking. Wiley & Sons, New York (1995)
9. Solis, F.J., Wets, R.J.: Minimization by random search techniques. Mathematics of operations research 6(1), 19–30 (1981)
10. Stouten, P.F.W., Frömmel, C., Nakamura, H., Sander, C.: An effective solvation term based on atomic occupancies for use in protein simulations. Molecular Simulation 10(2-6), 97–120 (1993)
11. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. on Evolutionary Computation 1(1), 67–82 (1997)

GA Inspired Heuristic for Uncapacitated Single Allocation Hub Location Problem

Vladimir Filipović, Jozef Kratica, Dušan Tošić, and Djordje Dugošija*

Abstract. In this article, the results achieved by applying GA-inspired heuristic on Uncapacitated Single Allocation Hub Location Problem (USAHLP) are discussed. Encoding scheme with two parts is implemented, with appropriate objective functions and modified genetic operators. The article presents several computational tests which have been conducted with ORLIB instances. Procedures described in related work round distance matrix elements to few digits, so rounding error is significant. Due to this fact, we developed exact total enumeration method for solving subproblem with fixed hubs, named Hub Median Single Allocation Problem (HMSAP). Computational tests demonstrate that GA-inspired heuristic reach all best solutions for USAHLP that are previously obtained and verified branch-and-bound method for HMSAP. Proposed heuristic successfully solved some instances that were unsolved before.

1 Introduction

The past four decades have witnessed an explosive growth in the field of network-based facility location modelling. The multitude of applications in practice is a major reason for the great interest in that field. Computer and

Vladimir Filipović · Dušan Tošić · Djordje Dugošija
University of Belgrade, Faculty of Mathematics, Studentski trg 16/IV, 11 000 Belgrade, Serbia
e-mail: vladaf@matf.bg.ac.yu, tdusan@mi.sanu.ac.yu, dugosija@matf.bg.ac.yu

Jozef Kratica
Mathematical Institute, Serbian Academy of Sciences and Arts, Kneza Mihajla 36/III, 11 001 Belgrade, Serbia
e-mail: jkratica@mi.sanu.ac.yu

* This research was partially supported by Serbian Ministry of Science under the grant no. 144007.

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 149–158.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

telecommunication networks, DHL-like services and postal networks, as well as transport systems can be analyzed as a hub network. All those systems contain a set of facilities (locations) that interact with each other, and with given distance and transportation cost. Instead of serving every user from its assigned facility with a direct link, hub network allows the transportation via specified hub facilities. Hubs serve as consolidation and connection points between two locations. Each node is allocated to one or more hubs and the flow from one node to another is realized via one or more hub facilities. Using switching points in the network and increasing transportation between them the capacity of the network can be used more efficiently. This strategy also provides lower transportation cost per unit.

There are various model formulations proposed for the problem of choosing subset of hubs in the given network. They involve capacity restrictions on the hubs, fixed cost, predetermined number of hubs and other aspects. Two allocation schemes in the network can be assumed: single allocation and multiple allocation concept.

In the single allocation hub location problem each node must be assigned to exactly one hub node so that all the transport from (to) each node goes only through its hub. Multiple allocation scheme allows each facility to communicate with more than one hub node. If the number of switching centers is fixed to p, we are dealing with p-hub problems. Capacitated versions of hub problems also exist in the literature, but the nature of capacities is different. The flows between hubs or between hubs and non-hubs can be limited. There are also variants of capacitated hub problems that consider limits on the flow into the hub node, through the node or fixed costs on hubs. One review of hub location problems and their classification can be found in [5, 6].

2 Mathematical Formulation

Consider a set $I = \{1, ..., n\}$ of n distinct nodes in the network, where each node denotes to origin/destination or potential hub location. The distance from node i to node j is C_{ij}, and triangle inequality may be assumed [6]. The transportation demand from location i to j is denoted with W_{ij}. Variable $X_{ik} = 1$ if the node i is allocated to hub established at node k. Therefore, $X_{jj} = 1 \Leftrightarrow j$ is hub. Otherwise, if node i is not allocated to hub at node k, variable $X_{ik} = 0$.

Each path from source to destination node consists of three components: transfer from an origin to the first hub, transfer between the hubs and finally distribution from the last hub to the destination location. In this, single allocation hub problem, is assumed that the flow from certain node involving only one hub node in all transportation. Parameters χ and δ denote unit costs for collection (communication from origin to the first hub) and distribution (communication from last hub to destination), while α represents transfer cost among hubs. The objective is location hub facilities to minimize the

total flow cost. The fixed cost for establishing hub node j is denoted with f_j. Using the notation mentioned above, the problem can be written as:

$$\min_{i,j,k,l \in N} W_{ij}(\chi C_{ik} X_{ik} + \alpha C_{kl} X_{ik} X_{jl} + \delta C_{jl} X_{jl}) + \sum_j X_{jj} f_j \quad (1)$$

subject to

$$\sum_k X_{ik} = 1, \quad for\ every\ i \quad (2)$$

$$X_{kk} - X_{ik} \geq 0 \quad for\ every\ i,\ k \quad (3)$$

$$X_{ik} \in \{0, 1\} \quad for\ every\ i,\ k \quad (4)$$

The objective function (1) minimizes the sum of the origin-hub, hub-hub and hub- destination flow costs multiplied with χ, α and δ factors respectively. Equation (2) forces single allocation scheme - each node is assigned to exactly one hub, and equation (3) allows that particular node can be assigned only to established hub.

3 Previous Work

Several methods for solving this problem are described in the literature [2, 3]. Due to the fact that this problem is NP hard, it is shown that its subproblem Hub Median Single Allocation Problem - HMSAP is NP hard [6], many authors recognized that good results can be obtained by applying evolutionary inspired solving strategies. In paper [1] several variants of hybridization Genetic algorithm and Tabu search are proposed. Obtained results are presented on CAB problem instances (from ORLIB, described in [4]). In paper, there isn't any result obtained by applying proposed hybrid algorithms on AP ORLIB instances.

Paper [11] proposed more advanced GA method for solving USAHLP. Proposed method uses more efficient representation, better initialization (initial number of hubs in item is set with more realism) and advanced crossover operator that is well suited to the problem domain. However, in this paper (it was also the case with previous methods) proposed GA method uses the simplest selection operator - proportional selection. Authors in [11] publish obtained results for both CAB and AP instances.

Paper [7] describes SATLUHLP heuristic that solves USAHLP. Heuristic SATLUHLP is hybrid of Simulated annealing and Tabu search. That heuristic is divided into three levels: the first level is to determine the number of hubs; the second level is to select the hub locations for a given number of hubs; and the third level is to allocate the non-hubs to the chosen hubs. Presented results are obtained by testing on ORLIB instances (CAB and AP) and those results are compared with [11].

4 Proposed GA Heuristic Method

Genetic algorithms (GAs) are problem-solving metaheuristic methods rooted in the mechanisms of evolution and natural genetics. The main idea was introduced by Holland [9], and in the last three decades GAs have emerged as effective, robust optimization and search methods.

4.1 Representation

Each gene of individual represents one node. Particular gene contains of two parts. The first part is 1 or 0 - it indicates if hub is established at corresponding node, or not. Second part contains number from set $\{0, 1, \ldots, n-1\}$. That number specifies which hub is assigned to fixed non-hub node. Naturally, every hub is assigned to itself. For instance, if non-hub node is assigned to the closest hub, then there will be 0 in the second part. Furthermore, if non-hub node is assigned to hub that is more distant to node than closest hub, but less distant than any other hub, there will be number 1 in second part of genetic code, etc.

The first part of genetic code is generated in random manner. Due to the fact that less distant hubs should be more often selected during generation, it is preferable that second part of genetic code contains large number of zeros. To accomplish that, probability that the first bit in each gene will be set to 1 is $1.0/n$. Bits that follows will have probability to be set to one equals to the half of its predecessor probability - e.g. $0.5/n$, $0.25/n$, $0.125/n$, ... respectively.

4.2 Objective Function

Fitness of the individual is calculated according to following procedure:

- First part of each gene gives indexes of established hubs.
- After set of established hubs is obtained, array of established hubs will be sorted (for each non-hub node) in ascending order, according to distance to that specific non-hub node.
- Element that corresponds to specific non-hub node is extracted from second part of every gene. If extracted element has value r (r = 0, 1, ..., $n-1$), then r-th element of (adequately) sorted array will be index of the hub which is specific node assigned to.
- Now, objective value (and fitness of individual) is obtained simply by summing distances source-hub, hub-hub and hub-destination, multiplied with load and with corresponding parameters χ , α and δ.

Sorting of established hubs array according to distance, for each individual, takes part in every generation and that requires the processor's extra work.

However, the obtained results confirm our estimate that the processor's extra work has very little influence on overall time of algorithm execution.

4.3 Genetic Operators

Genetic operators are designed in following way:

- GA uses FGTS [8] as selection operator. Parameter F_{tour}, that governs selection method is not changed during execution of GA, and its value is 5.4. That value is experimentally obtained. Moreover, FGTS selection with $F_{tour} = 5.4$ behave very well in solving some similar problems.
- After selection, one-position crossover operator has been applied. Probability of crossover is 0.85, which means that about 85% individuals in population will produce offsprings, but in approximately 15% cases crossover will not take part and offsprings will be identical to its parents. Crossover point is chosen on the gene boundary. Therefore, there is no gene splitting.
- Evolutionary method uses simple mutation, which pseudo randomly changes one bit in both parts of every gene. Mutation levels are different in different parts. The first bit in every gene mutates with probability $0.6/n$. The second bit in each gene mutates with probability $0.3/n$ and subsequent bits mutate with probability that is half of its predecessor mutation probability ($0.15/n$, $0.075/n$, $0.0375/n$, $0.01875/n$, etc.).

  During GA execution, sometimes happens that all individuals in population have same bit at specified position. Such bits are known as frozen bits. If number of frozen bits is l, then search space becomes 2^l times smaller and probability of premature convergence quickly rises. Selection and crossover can not change frozen bit. Probability of classic mutation is often too small to successfully restore lost subregions in search space. On the other side, if probability of classic mutation is significant, GA pretty much behave like pure random search procedure. Therefore, mutation probability will be increased only for frozen bits. In this GA, probability of mutation for frozen bits in first part is two and half times higher than probability for non-frozen bits, it is $1.5/n$. Probability of mutation for frozen bits in second part is one and half times higher comparing to non-frozen counterparts, so it is $0.225/n$, $0.1125/n$, $0.055625/n$, etc. Reason for lower mutation probabilities for bits in second part is importance that second part mainly contains zeros. In section that describes representation is already highlighted that zero in second part represents the nearest hub to specific non-hub node. Obtained experimental results also justifies probability setting that is described.

4.4 Other GA Aspects

There are many aspects (beside representation, objective function and genetic operators) that have significant influence on GA performance. Most important among them are:

- Population has 150 individuals. Number of individuals does not increase nor decrease during GA execution.
- GA uses steady-state replacement policy and elitist strategy - 100 best fitted individuals (e.g. elite individuals) are directly transferred into new generation and its fitness remains the same and should not be recalculated.
- Duplicate individuals are removed in every generation during GA execution. This is accomplished by setting fitness value of duplicated individual to zero, so that individual won't be selected to pass into new generation during selection phase. On that way, genetic diversity is preserved and premature convergence has very small probability.
- Sometimes, during GA execution it happens that individuals with the same fitness value but different genetic code dominate the population. If genetic codes of such dominating individuals are similar, it can bound GA execution to some local extremum. In order to avoid similar situations, we decided to limit the number of individuals with the same fitness and different genetic code. In the current implementation, that number is 40.
- GA execution is stopped after 1000 generations when larger instances are solved, or after 500 generations on small size USAHLP instances. Algorithm is also stopped if best individual does not improve its value during 200 generations.
- Furthermore, performance of GA is improved by cashing GA [10] and cash size is 5000.
- Previously described representation, initialization, selection and mutation prevent creation of incorrect individuals, so there is no need for some special correction.

5 Computational Results

Algorithms are tested on ORLIB instance set, taken from [4].

CAB (Civil Aeronautics Board) data set is based on information about civil air traffic among USA cities. It contains 60 instances, with up to 25 nodes and up to 4 hubs. In this instances is assumed that unit costs for collection and distribution (χ and δ) is 1. Results of the proposed GA implementation (just like implementations described in [7, 11]) obtain optimal solution for all instances, with extremely short execution times. Therefore, results on CAB instances are omitted from this paper and can be downloaded from `http://www.matf.bg.ac.yu/~vladaf/Science/USAHLP/cab.txt`.

Data for AP (Australian Post) set are obtained from Australian Post System. AP contains up to 200 nodes that represent postal areas. Smaller size AP instances are obtained by aggregation of the basic, large, data set. Distances among cities fulfill triangle inequality, but load is not symmetric at all. AP also includes fixed price for hub establishment. Suffix "L" in instance name will indicate that fixed costs are light, and suffix "T" will indicate heavy fixed costs. Larger AP instances, that are significantly larger and therefore more difficult, make that algorithm executes for longer time. Those instances will more likely give us better look on overall behavior of algorithm.

However, a new problem arises there: results (e.g. obtained solutions) that are described in paper [11] are sometimes significantly different to solutions that are obtained by proposed GA method. In direct, personal communication, we asked the author to help us to determine possible reasons for the observed differences. In his answer, Topcuoglu speculates that the reason for this is an accumulated rounding error, because he rounded the distance matrix to three decimal places. Results that are published in [7] are not completely identical to results that gives proposed GA method, but difference is much smaller comparing to [11], since distances are rounded up to six decimal places.

In order to completely clear up dilemmas, we decided to obtained exact solution of USAHLP subproblem, called Hub Median Single Allocation Problem - HMSAP. HMSAP problem is similar to USAHLP, but hubs are fixed. In other words, HMSAP should make assignment of hubs to non-hub nodes so overall traffic cost is minimal. Once when we get set of established hubs (note that all algorithms in comparison got the same set of established hubs), we obtained hub assignment by solving HMSAP problem with classical enumeration algorithm. It is widely known that such algorithm guaranties optimality of obtained solution.

Algorithms are executed on the computer with AMD Sempron 2.3+ processor, which works at 1578 MHz clock and have 256 MB memory. During experiments, computer works on UNIX (Knoppix 3.7) operating system. There were activated all C compiler optimizations during compiling, including AMD processor optimization. Proposed GA was executed 20 times for each problem instance.

Table 1 shows experimental results that are obtained by proposed GA method, results obtained by HMSAP and results presented in [7, 11]. First column identifies AP instance that is solved. Best solution obtained by GA is presented in column *GA.best*. Column t contains average time (expressed in seconds) that GA needs to obtain best solution, and column t_{tot} contains average time (also expressed in seconds) for finishing GA. In average, GA finishes after **gen** generations.

Quality of obtained solution is quantified as average gap (denoted as *avg.gap* and expressed in percents) and it is calculated by following formula:

$avg.gap = \frac{1}{20} \sum_{i=1}^{20} gap_i$, where gap_i represent gap that is obtained during i-th execution of GA on specific instance. Gap is calculated in respect to optimal solution $Opt.sol$ (if it is already known): $gap_i = \frac{sol_i - Opt.sol}{Opt.sol} 100$. In cases where optimal solution is not known in advance, gap is calculated in respect to best found solution $Best.sol$: $gap_i = \frac{sol_i - Best.sol}{Best.sol} 100$. Tables also contain standard deviation of the gap (denoted as σ) and it is calculated on the following way: $\sigma = \frac{1}{20} \sum_{i=1}^{20} (gap_i - avg.gap)^2$.

Column **Hubs** gives information about established hubs. Column **HMSAP Enu** contains information about obtained exact solution of the subproblem when hubs are fixed and next column contains time that enumeration algorithm spent in order to obtain solution. Columns **Topcu.** and **Chen.** contain Topcuoglu's and Cheng's results.

If there is abbreviation "n.t." in table cell, it means that problem instance is not tested. Abbreviation "n.n" means that solving was not necessary - for instance if there is only one established hub, assignment is trivial and it is not necessary to solve HMSAP problem. Abbreviation "n.f" means that algorithm did not finished its work, and "time" indicates that execution lasted more than one day.

All the cells in Table 1, where differences are so significant that can not be easily explained only by accumulation of rounding error, are bolded. There is also some chance that those differences are generated by some differences in downloaded AP problem instances, or by inadequate aggregation.

Papers [7, 11] present results only for AP instances with $\chi = 3$, $\alpha = 0.75$ and $\delta = 2$, so Table 1 contains data for direct comparison among proposed GA, Topcuoglu's algorithm and Cheng's algorithm. Results of proposed GA and HMSAP Enu algorithm on AP instances with different values for χ, α and δ can be downloaded from address `http://www.matf.bg.ac.yu/~vladaf/Science/USAHLP/ap.txt`.

Data in Table 1 indicate that, whenever exact enumeration algorithm obtain the solution, proposed GA also obtained the same solution. In some cases, for example 120T instance, when number of established hubs is small, HMSAP Enu finishes its work quicker than GA, but HMSAP Enu solves only subproblem with fixed established hubs. Furthermore, we can notice four cases where execution of enumeration algorithm lasted extremely long, but GA implementation obtained solution during very small amount of time.

GA implementation is also comparable in terms of running time with Topcuoglu and Chen methods. Note that running time of the GA increases at smaller rate than in [7, 11]. For example, for n=200, GA running time is approximately 20 seconds, while Topcuogly is 3000 seconds and Chen is 180 seconds.

Table 1 Results for comparison GA and HMSAP Enu on AP instances with $\chi = 3$, $\alpha = 0.75$, $\delta = 2$

Inst.	*GA.best*	*t*[*s*]	t_{tot}[*s*]	**gen**	*avg.gap*[%]	σ[%]	**Hubs**	**HMSAP Enu**	*t*[*s*]	**Topcu.**	**Chen.**
10L	224250.055	0.009	0.101	217	0.000	0.000	3,4,7	224250.055	0.04	224249.82	224250.06
10T	263399.943	0.020	0.113	249	0.000	0.000	4,5,10	**263399.943**	0.07	**263402.13**	263399.95
20L	234690.963	0.016	0.206	216	0.000	0.000	7,14	234690.963	0.11	234690.11	234690.96
20T	271128.176	0.029	0.213	229	0.909	1.271	7,19	**271128.176**	0.11	**263402.13**	271128.18
25L	236650.627	0.035	0.275	228	0.000	0.000	8,18	**236650.627**	0.20	**236649.69**	236650.63
25T	295667.835	0.004	0.233	201	0.000	0.000	13	n.t.	n.n.	**295670.39**	295667.84
40L	240986.233	0.101	0.554	244	0.221	0.235	14,28	**240986.233**	1.90	**240985.51**	**240986.24**
40T	293164.836	0.069	0.500	231	0.000	0.000	19	n.t.	n.n.	**293163.38**	293164.83
50L	237421.992	0.298	0.904	298	0.327	0.813	15,36	**237421.992**	4.16	**237420.69**	237421.99
50T	300420.993	0.008	0.592	201	0.000	0.000	24	n.t.	n.n.	**300420.87**	**300420.98**
60L	228007.900	0.415	1.205	306	0.546	0.919	18,41	228007.900	7.93	n.t.	n.t.
60T	246285.034	0.231	1.016	258	0.356	1.593	19,41	246285.034	7.54	n.t.	n.t.
70L	233154.289	0.451	1.489	286	0.000	0.000	19,52	233154.289	9.84	n.t.	n.t.
70T	252882.633	0.360	1.397	269	0.000	0.000	19,52	252882.633	9.88	n.t.	n.t.
80L	229240.373	1.143	2.397	383	1.143	1.286	22,55	229240.373	21.34	n.t.	n.t.
80T	274921.572	0.633	1.818	300	0.249	0.765	5,41,52	274921.572	4455	n.t.	n.t.
90L	231236.235	0.919	2.463	319	0.841	0.865	26,82	231236.235	87.23	n.t.	n.t.
90T	280755.459	0.437	1.934	257	0.133	0.395	5,41	280755.459	16.64	n.t.	n.t.
100L	238016.277	1.382	3.221	349	0.381	0.757	29,73	**238016.277**	69.69	**238017.53**	**238015.38**
100T	**305097.949**	0.365	2.180	239	0.000	0.000	52	**n.t.**	n.n.	**305101.07**	**305096.76**
110L	222704.770	3.025	5.205	478	1.430	1.611	32,77	222704.770	7 045	n.t.	n.t.
110T	227934.627	2.604	4.761	438	4.846	5.774	32,77	227934.627	6 718	n.t.	n.t.
120L	225801.362	2.304	4.775	384	0.392	0.778	32,85	225801.362	2 896	n.t.	n.t.
120T	232460.818	3.440	5.913	475	1.741	2.972	32,85	232460.818	2 934	n.t.	n.t.
130L	227884.626	3.563	6.661	428	1.098	1.037	36,88	n.f.	time	n.t.	n.t.
130T	234935.968	3.108	6.181	399	0.398	0.459	36,88	n.f.	time	n.t.	n.t.
200L	**233802.976**	11.521	19.630	482	0.398	0.815	43,148	**n.f.**	time	**228944.77**	**228944.18**
200T	**272188.113**	10.981	19.221	463	0.326	0.215	54,122	n.f.	time	**233537.93**	**233537.33**

6 Conclusions

In this article, we introduced a GA-inspired heuristic that solves the USAHLP by simultaneously finding the number of hubs, the location of hubs, and the assignment of nodes to the hubs. The assignment part (HMSAP) is successfully verified by the results of branch-and-bound method for all cases where exact HMSAP solution can be obtained in reasonable time.

In proposed method, two-part encoding of individuals and appropriate objective functions are used. Arranging located hubs in non-decreasing order of their distances from each non-hub node directs GA to promising search regions. We have used the idea of frozen bits to increase the diversity of genetic material by mutation. The caching technique additionally improves the computational performance of GA.

Extensive computational experiments indicate that the proposed method is very powerful and that the medium-size and large-size USAHLP instances can

be solved within a twenty seconds of computing time for sizes attaining 200 nodes. Such results imply that the GA may provide an efficient metaheuristic for real world USAHLP and related problems.

Hence, our future work could also concentrate on the speed-up of the algorithm by taking advantage of parallel computation and on GA hybridization with exact methods.

References

1. Abdinnour-Helm, S.: A hybrid heuristic for the uncapacitated hub location problem. European Journal of Operational Research 106, 489–499 (1998)
2. Abdinnour-Helm, S., Venkataramanan, M.A.: Solution Approaches to Hub Location Problems. Annals of Operations Research 78, 31–50 (1998)
3. Aykin, T.: Networking Policies for Hub-and-spoke Systems with Application to the Air Transportation System. Transportation Science 29, 201–221 (1995)
4. Beasley, J.E.: Obtaining test problems via internet. Journal of Global Optimization 8, 429–433 (1996), http://mscmga.ms.ic.ac.uk/info.html http://www.brunel.ac.uk/depts/ma/research/jeb/orlib
5. Campbell, J.F.: Hub Location and the p-hub Median Problem. Operations Research 44(6), 923–935 (1996)
6. Campbell, J.F., Ernst, A., Krishnamoorthy, M.: Hub Location Problems. In: Hamacher, H., Drezner, Z. (eds.) Location Theory: Applications and Theory, pp. 373–407. Springer, Heidelberg (2002)
7. Chen, F.H.: A hybrid heuristic for the uncapacitated single allocation hub location problem. OMEGA - The International Journal of Management Science 35, 211–220 (2007)
8. Filipović, V.: Fine-grained tournament selection operator in genetic algorithms. Computing and Informatics 22, 143–161 (2003)
9. Holland, J.: Adaptation in Natural and Artificial Systems. The University of Michigan Press (1975)
10. Kratica, J.: Improving performances of the genetic algorithm by caching. Computers and Artificial Intelligence 18, 271–283 (1999)
11. Topcuoglu, H., Court, F., Ermis, M., Yilmaz, G.: Solving the uncapacitated hub location problem using genetic algorithms. Computers & Operations Research 32, 967–984 (2005)

Evolutionary Constrained Design of Seismically Excited Buildings: Sensor Placement

Alireza Rowhanimanesh, Abbas Khajekaramodin,
and Mohammad-Reza Akbarzadeh-T.

Abstract. Appropriate sensor placement can strongly influence the control performance of structures. However, there is not yet a systematic method for sensor placement. In this paper, a general method based on a proposed constrained GA is suggested to optimally place sensors in structures. The optimal placement scheme is general for passive, active and semi-active controls and it does not depend on the control strategy and nonlinear dynamics of the control system. Due to low cost, high reliability, control effectiveness as well as installation simplicity, acceleration type sensors are considered. The proposed method is applicable to new or existing buildings, as the accelerometer placement is trivial. The efficiency of proposed method is evaluated on the benchmark building for placement of 5 and 3 sensors. The results show that the performance of control system with 5 and even 3 optimally placed sensors is at least 8% better than the original benchmark building design with 5 sensors. Generally, the proposed method is a simple and efficient practical approach to achieve improved performance using the fewest number of instruments.

1 Introduction

Seismic loads such as earthquakes and strong winds might excite the civil structures and create sever damages. Thus, designing a system that can protect and control structures against these natural hazards is valuable. The first proactive research on control of structures began by Yao in early 1970s [1]. From its beginning to present, three major control schemes including passive, active and semi-active were introduced, among which the semi-active approach is the most efficient and recent. From a control paradigm perspective, these control approaches can also be divided into conventional and intelligent approaches. Prior research by authors such as in Rowhanimanesh et al. (2007) [2-3], Khajekaramodin et al. (2007) [4], Akbarzadeh-T and

Alireza Rowhanimanesh · Mohammad-Reza Akbarzadeh-T.
Cognitive Computing Lab, Department of Electrical Engineering,
Ferdowsi University of Mashhad, Mashhad, Iran
e-mail: rowhanimanesh@ieee.org, akbarzadeh@ieee.org

Abbas Khajekaramodin
Department of Civil Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
e-mail: karamodin@gmail.com

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 159 – 169.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

Khorsand (2005, 2006) [5-7] and Rezaei, Akbarzadeh-T et al. (2003) [8] can be placed in the latter category.

One of the challenging problems in the area of structural control is the optimal placement of actuators and sensors that is often done through an ad hoc process, without any systematic method. However, appropriate placement can strongly influence the performance of control system and significantly minimize the costs. In most of control system designs, acceleration sensors are used with respect to their worthy practical potentials. In comparison with actuators, sensors especially accelerometers can be relocated easily after design and construction of structure. Thus, optimal placement of sensors, that is the main topic of this paper, could be applied for all constructed or under-construction structures to improve the performance and efficiency of control system. Generally, optimal placement of sensors and actuators is an approach to reach to the best performance using the least of instruments, thus it could reduce the costs and increase the efficiency of structural control system. Some of the previous works are briefly reviewed in continue.

Martin et al. (1980) [9] considered the placement of active devices in structures for modal control. Lindberg at al. (1984) [10] discussed the appropriate number and placement of devices based on modal control. Vander Velde et al. (1984) [11] focused on structural failure modes to place the devices, and some other works performed by Ibidapo (1985) [12] and Cheng et al. (1988) [13]. Most of these methods are often problem specific. Takewaki (1997) [14] used a gradient-based approach to search for an optimal placement. Teng et al. (1992) [15] and Xu et al. (2002) [16] developed an incremental algorithm and Wu et al. (1997) [17] used both iterative and sequential approaches. Zhang et al. (1992) [18] and Lopez et al. (2002) [19] proposed sequential search methods for optimal damper placement. The main problem of these methods is converging to local optima. Simulated annealing as a guided random search was employed to place devices by Chen (1991) [20] and Liu (1997) [21]. Although these methods could solve the problem of local optima, they couldn't always provide general and efficient techniques. Finally, the most recent approaches have focused on evolutionary algorithms such as genetic algorithms (GA). Simpson et al. (1996) [22] used GAs to optimize the placement of actuators for a specific active control problem. Ponslet et al. (1996) [23] employed a GA approach to design an isolation system. Abdullah (2001) [24] combined GA and a gradient-based optimization technique for optimal placement of controllers in a direct velocity feedback control system. Li (2001) [25] have developed a multilevel GA to solve a multitasking optimization problem. Singh et al. (2002) [26] considered the placement of passive devices in a multistory building. Wongprasert et al. (2004) [27] employed GA for identifying the optimal damper distribution to control the nonlinear seismic response of a 20- story benchmark building. Tan (2005) [28] used GA to propose an integrated method for device placement and active controller design. Akbarzadeh-T (the third author) and Khorsand (2005, 2006) [5-7] applied meta-level multi-objective GA as well as evolutionary quantum algorithms in structural design. Cimellaro et al. (2007) [29] considered the optimal placement of dampers via sequential search and concepts of optimal control theory. Authors in 2007 [2-3] placed the actuators optimally on

the 20-story benchmark building using a new constrained GA. Beal et al. (2008) [30] placed sensors optimally for structural health monitoring using generalized mixed variable pattern search (MVPS) algorithm.

Regarding the sensor placement problem, the constraints are often simpler than actuator placement. In [2-3], we have proposed a new constrained GA to handle the constraints using a new interpretation function, but the computational load of the constraint handling method is too complex and not efficient for sensor placement since simpler and faster methods are expected. As discussed in section 2.2, undoubtedly the best method for constraints handling is direct approach using preserved operators which is much more efficient and faster than indirect constraint handling. But in most cases such as actuator placement problem [2-3], finding preserved operators is too difficult or even impossible such that we have to use indirect approach. In contrast, in this paper we propose very simple preserved crossover and mutation operators to provide much simpler, faster and more efficient approach to sensor placement. To evaluate the efficiency of the proposed approach, we apply the benchmark building introduced by Ohtori et al. in 2004 [34]. Many other papers have also evaluated their results on this benchmark building [2-4, 27 and 36]. The efficiency of the proposed method is evaluated on 5 and 3 sensor placement tasks in a 20 story building. The results indicate that the performance of control system with 5 and even 3 optimally placed sensors is at least 8% better than the performance of original benchmark building design with 5 sensors.

2 Optimal Placement of Sensors

2.1 Defining the Optimization Problem

From the optimization perspective, optimal placement of sensors is a constrained, nonlinear and binary optimization problem. These difficulties are explained in two parts. A) The problem is binary and constrained. A decision variable is defined as the availability of sensor in a given floor (a binary variable), 1 means a sensor is located on the given floor and 0 means that there is not any measurement there. For example for a 20-story building, there are 20 decision variables. As mentioned before, regarding the practical advantages of accelerometers, in this paper, we suppose that only acceleration of some of the stories is measured. With respect to the practical and economical considerations, designers often try to achieve the desired performance by measuring only the accelerations of limited number of floors. Thus, in practice the total number of sensors is restricted which is a linear constraint. Generally, these constraints are the most common ones for sensor placement which are shown in Table 1. The proposed method can successfully deal with these constraints, however if there are different linear or nonlinear constraints including equalities or inequalities, the proposed method can be simply modified. B) The problem is nonlinear, discrete and not analytical. The analytical relation could not be extracted simply and precisely between the configuration of sensors as an input and a control objective as an output. The input space is discrete; the objective functions often use the operators such as maximum or absolute

value that they are not analytical. The model of the structure is a system of nonlinear differential equations that must often be solved numerically. Furthermore, because of complexity and nonlinearity of the problem, the method must be able to seek the global optima. For these reasons, derivative-based and local optimizers can not be performed. In this paper, we propose constrained GA to solve this problem since handling constraints in GA is a big challenge.

Table 1 Constraints for Optimal Sensor Placement

1	X_i: binary {0,1}
2	$X_1+ X_2+\ldots +X_n=N_{tot}$

2.2 Constrained GA

As mentioned in [31-33], there are two major approaches for handling constraints in GA: A. Direct constraint handling which contains four methods: 1.eliminating infeasible candidates, 2. repairing infeasible candidates, 3. preserving feasibility by special operators, 4. decoding such as transforming the search space. Direct constraint handling has two advantages. It might perform very well, and might naturally accommodate existing heuristics. Because the technique of direct constraint handling is usually problem dependent, it has several disadvantages. Designing a method for a given problem may be difficult, computationally expensive and sometimes impossible. B. Indirect constraint handling: Indirect constraint handling incorporates constraints into a fitness function. Advantages of indirect constraint handling are: generality (problem independent penalty functions), reducing problem to 'simple' optimization, and allowing user preferences by selecting suitable weights. One of disadvantages of indirect constraint handling is loss of information by packing everything in a single number and generally in case of constrained optimization, it is reported to be weak. Generally, one of the best approaches for handling constraints in GA is using preserved operators that create feasible solutions, this means that with a feasible initial population, GA could search on the feasible space. A difficulty is finding some suitable and efficient preserved operators that sometimes it is nearly impossible.

2.3 Proposed Constrained GA

In this paper, an innovative constrained GA is proposed based on new preserved crossover and mutation operators to handle the constraints of Table 1 efficiently and very simply. The optimal scheme provides the third type of direct constraint handling. The structure of proposed constrained GA is presented in Figure 1. Each gene determines the availability of sensor in a given floor, 1 means a sensor is located on the given floor and 0 means that there is not any measurement there. So, the length of a chromosome is equal to the number of stories of the building. The type of GA is binary. The initial population is random and feasible. It means that each chromosome satisfies the constraints of Table 1. According to Figure 1, the fitness evaluation contains 3 stages. Regarding the given chromosome which determines the

specific placement of sensors, first an LQG controller is designed. Then the closed loop control system is excited by 10 seconds of El Centro earthquake. The response of the controlled structure as well as the value of cost function is calculated. This cost determines the fitness of the given chromosome. After selection, the preserved crossover and mutation create the next feasible generation, therefore GA always searches in feasible space. The proposed preserved crossover consists of 4 levels. First, two chromosomes are selected as parents. Regarding any given pair, there are 4 types of cylinders. The proposed crossover just works on two types 1-0 and 0-1. According to Figure 1, the 1-0 and 0-1 cylinders are determined. Then 50% of each type is randomly selected and in continue the genes are exchanged between two parents of each selected cylinder. As you can see, the mentioned operations keep the feasibility of the chromosomes. The result is two feasible offspring chromosomes. Regarding Figure 1, the proposed preserved mutation is really simple. A 1-0 or 0-1 pair is randomly selected from the chromosomes of a population. Then 1 is switched to 0 and 0 is switched to 1. This operation keeps the feasibility of the product. The suggested preserved operators are very simple and efficient. The method is general for all types of structural control problems and it is independent from the dynamics and strategies of control system.

3 Benchmark Building

To evaluate the efficiency of the proposed method, a 20-story benchmark building introduced in 2004 [34] has been used. The benchmark building definition paper has introduced three typical steel structures, 3-, 9-, and 20-story. The main purpose of introducing benchmark buildings is to provide a basis for evaluating the efficiency of various structural control strategies. Hence, numerical model of benchmark structures as well as Matlab files have been mentioned and discussed in the paper [34]. Also, 17 criteria with 4 historical earthquakes have been introduced for evaluation. Furthermore, practical constrains and conditions of control system have been stated such as sensor noise, sample rate, A/D and D/A characteristics, capacity of actuators and etc. Finally, a sample LQG active control system has been designed based on acceleration feedbacks.

3.1 Sample LQG Control System

In the benchmark paper, the LQG controller has been designed based on reduced-order linear model of 20-story nonlinear benchmark structure. The 25 active actuators are placed and 5 feedback measurements are provided by accelerometers located at some various locations on the structure. The active actuators are assumed to be ideal and the dynamics of them are neglected. In the sample control system of benchmark paper, the placements of sensors and actuators are not optimal. The acceleration sensors are located on stories 4, 8,12,16,20 that are shown in Figure 3. As it is indicated in the following, this distribution is not optimal. In this paper, the proposed method is performed on the benchmark paper sample control system

to achieve the optimal placement of sensors. The process is shown in Figure 1. All the conditions are as same as the benchmark paper and only the placement of sensors is optimized. The results demonstrate that using proposed method, better performance of the sample control system could be achieved only with optimal configuration of 5 sensors. Next, the proposed method is applied for placement of 3 acceleration sensors. The results show that the performance of control system with 3 optimally placed sensors is 8% better than the performance of benchmark sample control system with 5 sensors.

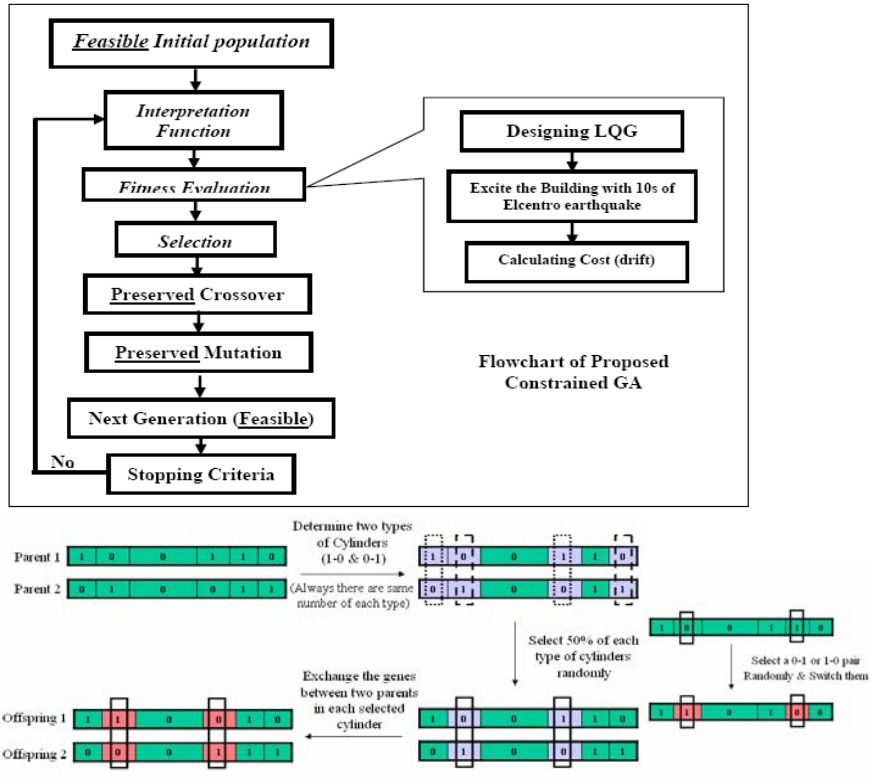


Fig. 1 Proposed constrained GA (top), Proposed preserved crossover (middle), Proposed preserved mutation (bottom) (N_{tot}=3)

4 Simulation Results

The following figures show the results of evaluation. First, the proposed method is applied to place 5 acceleration sensors optimally (Figure 3). The characteristics of GA are mentioned in Table 2. The proposed method used the first 10

seconds of El Centro earthquake that is a far-field earthquake. The cost function is MPA (maximum of peak of absolute value) of drifts among 20 stories when the benchmark building is excited by the first 10s of El Centro earthquake. In this study, the objective is minimization of story drifts. Other objectives such as acceleration or hybrids can be used. Moreover it is better to perform both near-field (like Kobe) and far-field (like El Centro) earthquakes. Figure 2 shows the convergence of the proposed evolutionary method. Next, the optimal solution was tested on 50s of two far-field (El Centro, Hachinohe) and two near-field (Northridge, Kobe) historical earthquakes.

Table 2 Characteristics of GA which is used in the simulation

Type of GA	Binary , Single-objective
Mutation rate	Fixed 0.3, New Preserved Mutation
Crossover	New Preserved Crossover
Population size	20
Selection	Keep 50% , give high probability to better individuals
Elitism	1 elite
Initial Population	Random , Feasible
Objective	Minimizing max of peak of abs of drifts

The optimal placement scheme has also been performed for placement of 3 sensors (Figure 3). The results were compared and indicate that the performance of control system with 3 optimally placed sensors is 8% better than the performance of benchmark paper's sample control system with 5 sensors. The results in Figure 4 demonstrate that using proposed method higher degree of performance could be achieved from the existing sensors without adding any extra instrumentation. Also, as a matter of fact, the proposed method is a simple and efficient approach to achieve the best performance using the least of instruments.

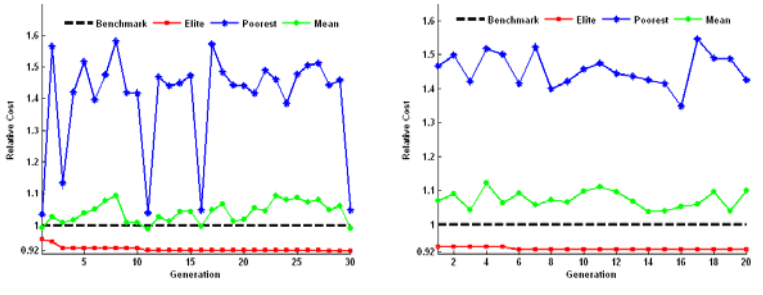


Fig. 2 Convergence of the Proposed Constrained GA – (relative cost = MPA of proposed placement / MPA of benchmark placement), Left: 5 sensors (optimal cost: 0.9171), Right: 3 sensors (optimal cost: 0.9259)

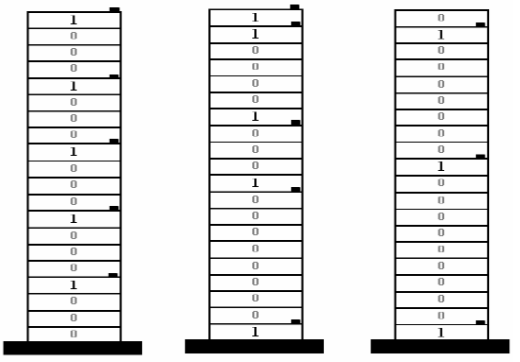


Fig. 3 Optimal placement of sensors: proposed method vs. benchmark paper

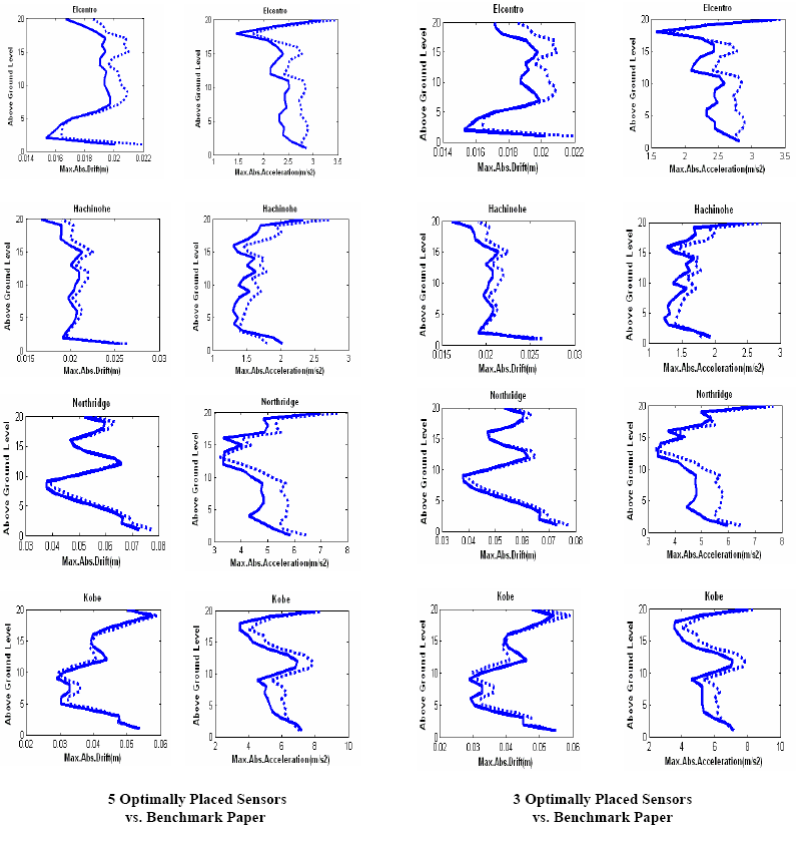


Fig. 4 Benchmark paper (dotted) vs. proposed method (solid): The results of evaluating the optimal placement scheme using 4 historical benchmark earthquakes

5 Conclusion

In this paper, we propose a general evolutionary approach for optimal sensor placement with handling the practical constraints. To handle constraints, we propose new preserved GA operators. The method is simple, efficient and flexible enough for passive, active and semi-active structural controls with any intelligent or conventional control strategy. It also allows the designer to consider all dynamics of control system. The results indicate the success of the method to improve the performance of the benchmark sample control system. In this work, we just considered minimization of drifts as the objective and only a far-field earthquake, El Centro, was used. For real designs, the designer is highly recommended to consider minimization of both drift and acceleration versus at least one near-field and one far-field earthquakes. In fact, the proposed approach is a simple and efficient way to achieve the higher performance of control system using the fewer instruments, lower cost and without adding any extra device.

References

1. Yao, J.T.P.: Concept of structural control. ASCE J. Stru. Div. 98, 1567–1574 (1972)
2. Rowhanimanesh, A., Khajekaramodin, A., Akbarzadeh, T.M.-R.: Evolutionary constrained design of seismically excited buildings, actuators placement. In: Proc. of the First Joint Congress on Intelligent and Fuzzy Systems, ISFS 2007, Mashhad, Iran, August 29-31 (2007)
3. Rowhanimanesh, A.: Intelligent control of earthquake-excited structures, B.Sc. Thesis, Department of Electrical Engineering, Ferdowsi University of Mashhad, Iran (2007)
4. Khajekaramodin, A., Haji-kazemi, H., Rowhanimanesh, A., Akbarzadeh, T.M.-R.: Semi-active control of structures using neuro-inverse model of MR dampers. In: Proc. of ISFS 2007 (2007)
5. Khorsand, A.-R., Akbarzadeh, T.M.-R.: Multi-objective Meta Level GA for Structural Design. accepted for publication at the Journal of Franklin Institute (March 2006)
6. Akbarzadeh, T.M.-R., Khorsand, A.-R.: Evolutionary Quantum Algorithms for Structural Design. In: Proc. of the 2005 IEEE Conf. on Systems, Man and Cybernetics, pp. 3077–3081 (2005)
7. Akbarzadeh, T.M.-R., Khorsand, A.-R.: Multi-objective Meta Level Soft Computing Based Evolutionary Design of Large-Scale Structures. In: Proc. of the 2005 World Congress on Fuzzy Logic, Soft Computing and Computational Intelligence, IFSA, pp. 1286–1293 (2005)
8. Rezaei-Pajand, M., Akbarzadeh, T.M.-R., Nikdel, A.: Active Structure Control by Online Earthquake Movement Prediction using Neural Networks. In: Proceedings of the Conference on Intelligent Systems (in Farsi), Mashhad, Iran, October 14-16, pp. 651–661 (2003)
9. Martin, Soong: Modal control of multistory structures. Journal of Engineering Mechanics, 613–623 (1980)
10. Lindberg Jr., R.E., Longman, R.W.: On the number and placement of actuators for independent modal space control. J. Guid. Control Dyn., 215–221 (1984)
11. Vander Velde, W.E., Carignan, C.R.: Number and placement of control system components considering possible failures. J. Guid. Control Dyn. 7(6), 703–709 (1984)

12. Ibidapo, O.: Optimal actuators placement for the active control of flexible structures. J. Math. Anal. Appl., 12–25 (1985)
13. Cheng, F.Y., Pantelides, C.P.: Optimal placement of actuators for structural control. Technical Rep. No. NCEER-88-0037, Multidisciplinary Center for Earthquake Engineering Research, Buffalo, N.Y. (1988)
14. Takewaki, I.: Optimal damper placement for minimum transfer function. Earthquake Eng. Struct. Dyn., 1113–1124 (1997)
15. Teng, J., Liu, J.: Control devices optimal installment for the new vibration control system of multi-structure connection. In: Proc. 10th WCEE, pp. 2069–2072 (1992)
16. Xu, Y.L., Teng, J.: Optimal design of active/passive control devices for tall buildings under earthquake excitation. Struct. Des.Tall Build., 109–127 (2002)
17. Wu, B., Ou, J.-P., Soong, T.T.: Optimal placement of energy dissipation devices for three dimensional structures. Eng. Struct. 19, 113–125 (1997)
18. Zhang, R.H., Soong, T.T.: Seismic design of viscoelastic dampers for structural application. J. Struct. Eng., 1375–1392 (1992)
19. Lopez, G., Soong, T.T.: Efficiency of a simple approach to damper allocation in MDOF structures. J. Struct. Control, 19–30 (2002)
20. Chen, G.S., Bruno, R.J., Salama, M.: Optimal placement of active/passive control of flexible structures. AIAA, 1327–1334 (1991)
21. Liu, X.J., Begg, D.W., Matravers, D.R.: Optimal topology/actuator placement design of structures using SA. J. Aerosp. Eng., 119–125 (1997)
22. Simpson, M.T., Hansen, C.H.: Use of genetic algorithms to optimize vibration actuator placement or active control of harmonic interior noise in a cylinder with floor structure. Noise Control Eng., 169–184 (1996)
23. Ponslet, E.R., Eldred, M.S.: Discrete optimization of isolator locations for vibration isolation system. In: Proc. AIAA, NASA, and ISSMO Symp. on Multi-Disciplinary Analysis and Optimization,Technical Papers, Part II, pp. 1703–1716 (1996)
24. Abdullah, M.M., Richardson, A., Hanif, J.: Placement of sensors and actuators on civil structures using genetic algorithms. J. Earthquake Eng. Struct. Dyn., 1167–1184 (2001)
25. Li, Q.S., Liu, D.K., et al.: Multi-level design model and genetic algorithm for structural control system optimization. J. of Earthquake Eng. and Structural Dynamics, 927–942 (2001)
26. Singh, M.P., Moreschi, L.M.: Optimal placement of dampers for passive response control. Earthquake Eng. Struct. Dyn. 31, 955–976 (2002)
27. Wongprasert, N., Symans, M.D.: Application of a genetic algorithm for optimal damper distribution within the nonlinear seismic benchmark building. J. Eng. Mech., 401–406 (April 2004)
28. Tan, P., Dyke, S.J., et al.: Integrated Device Placement and Control Design in Civil Structures Using Genetic Algorithms. J. of Structural Eng., 1489–1496 (October 2005)
29. Cimellaro, G.P., Retamales, R.: Optimal softening and damping design for buildings. J. of Structural Control and Health Monitoring, V 14(6), 831–857 (2007)
30. Beal, J.M., Shukla, A., Brezhneva, O.A., Abramson, M.A.: Optimal sensor placement for enhancing sensitivity to change in stiffness for structural health monitoring. Journal of Optimization and Engineering 9(2), 119–142 (2008)
31. Craenen, B.G.W., Eiben, A.E., van Hemert, J.I.: Comparing Evolutionary Algorithms on Binary Constraint Satisfaction Problems
32. Santos, A., Dourad, A.: Constrained GA applied to Production and Energy Management of a Pulp and Paper Mill

33. Schoenauer, M., Xanthakis, S.: Constrained GA optimization. In: Proceedings of the 5th International Conference on Genetic Algorithms, Urbana Champaign (1993)
34. Ohtori, Y., Christenson, R.E., Spencer Jr., B.F., Dyke, S.J.: Benchmark Control Problems for Seismically Excited Nonlinear Buildings. J. of Eng. Mechanics, 366–385 (April 2004)
35. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms. John Wiley & Sons, Inc., Chichester (2004)
36. Kim, et al.: Optimal Neurocontroller for Nonlinear Benchmark Structure. J. of Eng. Mechanics (2004)

Applying Evolution Computation Model to the Development and Transition of Virtual Community under Web2.0

Huang Chien-hsun and Tsai Pai-yung

Abstract. Under Web2.0, everyone is the content producer and consumer concurrently. The users' contribution to the environment is the key element to enrich the content. The connection among the individuals forms the social network, and become the virtual community which is socially open to every individual who wants to share information and join the community. The anticipation of the web companies in the open environment of Web2.0 is to have a bigger-and-better community. However, in the present phase of development, there are still many questions remaining, such as, will the virtual communities be expanded like the anticipation? Is the inference correct? In this paper, we propose a novel method to conduct the research based on the Particle Swarm Optimization (PSO). By simulating the behaviors of the individuals in the communities, we are able to demonstrate that there's a limitation of the community development in the Web 2.0 environment.

1 Introduction

The concept of "Web 2.0" is proposed by Tim O'Reilly [7]. The conclusion of the article is that "Network effects from user contributions are the key to market dominance in the Web 2.0 eras." In other words, Web 2.0 is not only a technology, but also an attitude [10]. It's about enabling and encouraging participation through "open" applications and services. The term "open" not only means technically open with appropriate APIs, but also emphasizes the importance of socially open, with rights granted to use the content in new and exciting context. Paul Miller [6] attempts to extract some of the important principles about Web 2.0.

The social network in the Web2.0 community is established based on interests and benefits. In other words, computer networks are social network [13]. Strong relationships as well as weak ties within people exist in the networks. It is especially useful for maintaining contact with "weak relationship" persons and group. Because weak ties are more socially heterogeneous than strong ties, they connect people to diverse social milieus and provide a wider range of information [3].

Huang Chien-hsun · Tsai Pai-yung
Institute of Information Management, National Chiao Tung University, No. 1001, Ta Hsueh Road, Hsinchu 300, Taiwan, ROC
e-mail: {katwin.huang, halohowau}@gmail.com

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 171 – 180.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

When the community is established, in order to attract new members to join this community and gather more information, the users try to build connections with other person by providing content in the initial phase. The state of the individuals will change over time when interaction built. And will the size of virtual society under web2.0 tend to unlimitedly growing? In fact, the link in the community needs physical cost. When the users can't afford the cost, the community development is constraint limiting. The community wouldn't expand scale-free. [14] The relationships between the individuals would become too much burden to sustain or continually expand. So how to make a decision about lasting relationship is a phenomenon we might want to observe.

Being affected by the whole society, the individuals will make human society decision. Our study applied PSO (Particle Swarm Optimization) to refer to individuals' human society decision. By using the PSO methodology, we propose a novel way to conduct this research. Starting from the aspect of users, we consider the human society decision to probe the evolution of the communities in the Web2.0 environment. By simulating the behavior of individuals in the communities, we try to observe the change of the communities through time. The experiment result shows that there's a limitation of the community development in the Web 2.0 environment.

The outline of this paper is as follows. In Section 2, we review the related work in the existing literature to orient the readers on the main idea of this paper. In Section 3, we describe the proposed simulation process of the experiment. Section 4 presents experiment result. The final section draws the conclusion of this paper.

2 Related Work

To orient the readers on the main concepts of this paper, we will review the following key concepts: social network and virtual community, particle swarm optimization (PSO) and the behavior of community through PSO methodology.

2.1 Social Network and Virtual Community

Social network [1] is social architecture constructed by many vertices which usually represent the individual or organization, and the social network represents the varying kinds of social relationships. It connects the individuals and the organizations through different social relationships. By designing a simulation experiment, we can analyze the interaction and effects between individuals. When computer networks connect people and organizations, they are the infrastructure of social network. We can infer the link between individuals as social network on the internet, which Rheingold [8] describes it as "Virtual Community."

Stanley Milgram [5] proposed the general idea about Six Degrees of Separation. The formal Six Degrees of Separation theory [4] is a description of the Small-world phenomenon in the society. For all these networks, there are constraints limiting the addition of new links [14], and there are two classes of factors to address this viewpoint. The first one is about aging of the vertices. This fact implies that even a very highly connected vertex will, eventually, stop receiving new links. The vertex is still part of the network and contributing to network

statistics, but it no longer receives links. Second, the other factor is the cost of adding links to the vertices or the limited capacity of a vertex.

But why people intend to accept this concept? The Small-world phenomenon does exist, just like Kahneman and Tversky propose that concept of availability heuristic [12]. People tend to accept the experience that they had been through. Rather than living in a "small, small world," we may live in a world that looks like a bowl of lumpy oatmeal, with many small worlds loosely connected and perhaps some small worlds not connected at all [5]. Just like the "lumpy oatmeal" phenomenon, six could be a big number.

In the social network, the probability of forming the link with every individual is not equal. The neighbor is easier to get to know each other. So in our simulation, the relationship of direct link only exists among individual and its neighboring vertices in the scope of its energy called the first layer. And the link other than the first layer forms the indirect link through the vertices of first layer.

In our study, on the internet environment, the concept of community not only exists in the specified area, but also exists based on the same hobby and interest. The interaction and the connection of individuals are established to form the virtual community.

2.2 Particle Swarm Intelligence

The society is a combination of individuals, or we can call "swarm." The interaction between the individual and the environment or between the individuals, which uses the bounded information to develop dynamic group behaviors, forms the swarm intelligence. In the computational intelligence area, there're algorithms been proposed based on swarm intelligence, and Particle Swarm Optimization (PSO) is one of them. Particle Swarm Optimization (PSO) is a technology of evolutionary computation created by Eberhart and Kennedy [11]. The concept of PSO, which comes from the analogy with the society which is similar to genetic algorithm, is a kind of tool based on the evolution of iterations.

An important feature of PSO is that the particles have their own memory. The information-sharing mechanism is different from Genetic Algorithm (GA). PSO initialize a swarm of random particles, finding the optimum toward the step of iteration. Particles adapt themselves toward following two "Best Value" in each iteration. One is the current optimum found by particle itself called pBest, another one is the current optimum found by the whole swarm called gBest.

When finding the two "Best Value", the particles adapt their current velocity and position based on following formula:

$$\mathrm{V}_{new} = Weight \times V_{current} + c_1 \times RV \times \left(P_{pBest} - P_{present}\right) + c_2 \times RV \times \left(P_{gBest} - P_{present}\right) \quad (1)$$

$$P_{new} = P_{present} + \mathrm{V}_{new} \quad (2)$$

In equation (1), V is the velocity of the particle; P is the position of the particle. V_{new} is the new velocity of the particle after adaption; $V_{current}$ is the current velocity of the particle. In equation (2), P_{new} is the new position of the particle after adaption; $P_{present}$ is the current position of the particle. P_{pBest} is the current optimum

found by particle itself. P_{gBest} is the current optimum found by the whole swarm. Learning factors, c_1 and c_2, are usually 2 [11]. RV is a random number between 0 and 1.

Akerlof [2] discusses social decision. It describes the behavior of individual making choices, being affected by the social element, will refer to the decision of the group (other individuals). The interaction between two individuals or the individual and the environment is an unique model, and we can sum up as a regular pattern of group behavior [9]. For example, before buying a cell phone, we always try to find the user experience in the community or other related information. Meanwhile, we will share our experience and opinion to the community.

The development and transition of virtual community somehow is a kind of social decision. The process of social decision is similar to PSO, especially the feature of memory. The PSO technology will have a better performance in solving our problem, thus we apply PSO to present our concept.

3 Proposed Model

3.1 Initia1 Individual

In the society, the meaning of community is group, which formed by individuals because of the small geographical distance or the frequent interaction. By creating a 2-axis graph to plot the movement of individuals, we try to simulate the virtual distance and the interaction between individuals on the internet.

The second step is Initial Individuals, and we try to simulate one hundred individuals under the assumption 1 of this project. In this step, we define the initial *energy* of the individual, and construct the *characteristic* attributes. After plotting the individuals on the plane, we record the initial status.

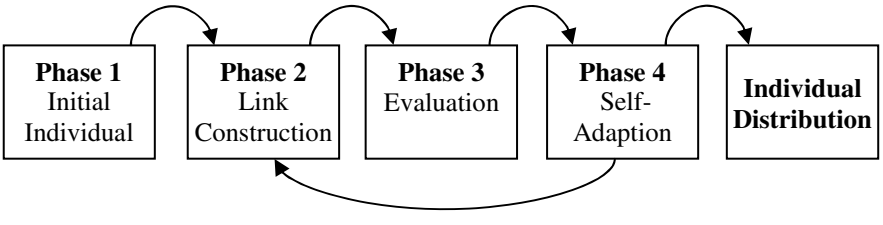


Fig. 1 We propose a framework to simulate human behavior, and this shows the implement procedure

The assumption of this project is as follows:

- ˜ Assumption 1: The energy of the individuals is lognormal-distributed. Because most individuals are willing to interact with other individual at first, in other words, it means they are full of passions to build connections.

- ˜Assumption 2: The position of the individuals is uniform-distributed on the plane. The individuals are spread on the 2-axis graph in a random way.
- ˜Assumption 3: The character of the individuals is uniform-distributed between 0 and 1.

In the experiment, we set up two characters: one affects the position of the individual, and the other one affects the level of activeness (energy).

3.2 Link Construction

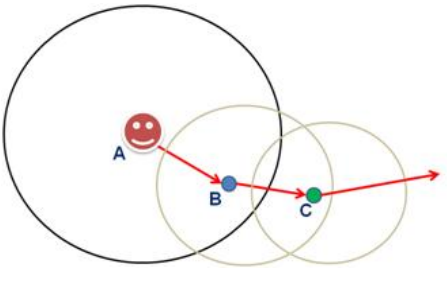
In the phase 2, we conduct the link construction. The social interaction between the individuals is built, and the relationship is constructed on the internet. We can calculate the cost of linkage. The definition of link is as follows:

1. Direct Link (A to B): In A's view point, when B is located in the A's scope of energy, a link will be established from A to B, if the distance from A to B is less than the radius of A. There is a direct link between A=(a_x,a_y)and B=(b_x,b_y), and the distance is the *Euclidean distance* from A to B:

$$\sqrt{\left(a_x - a_y\right)^2 + \left(b_x - b_y\right)^2} \tag{3}$$

   Constructing a relationship between two individuals implies that the individuals can communicate with each other directly, and the relationship between them is relatively strong than indirect link.
2. Indirect Link (A to C): The link between two individuals is established by more than two links. This kind of relationship is always established based on trust. Individuals are willing to build connection with the friends of their friends. The relationship between indirect individuals could be built with lower energy. According to the theory of Six Degrees of Separation, the maximum number of links between two individuals is six. There is cost between individuals when the link is constructed. $Cost_{XY}$ is the cost of indirect link from *X* to *Y*; *X* and *Y* are the individuals. r_x is the radius of *x*.

Fig. 2 Linkage Layer (Friend of a friend)



$$Cost_{AC} = Cost_{AB} + \left(1 - \frac{Cost_{AB}}{r_A}\right) \times Cost_{BC} \tag{4}$$

3.3 Evaluation

In this phase, we compute the fitness value, and try to figure out the best decision of the individual. Here we adopt the PSO (Particle Swarm Optimization) methodology to decide the best strategy of the individual.

We regard fitness value as the individual's status in current position and energy. When building connections, the number of links is the cost to the individuals. Individuals will feel uncomfortable, which results in changing fitness value, we calculate it as follow:

$$\text{Fitness Value (Comfortable Degree)} = \text{Energy} - \text{Total Cost}. \tag{5}$$

We obtain the fitness value (comfortable) for each individual. If the fitness value of the individual is bigger than the fitness value of *pBest*, we will replace the original *pBest* with current individual and determine the *gBes*t in each iteration.

3.4 Self-adaption

An individual is not comfortable in present status, when the fitness value is smaller than 0. He will try to make himself comfortable by self-adaption. There are two strategies that can be chosen in this phase, and the most part of the individuals is accompanied with two strategies.

1. Position-changing strategy:

The individual could change his position to make him more comfortable. We use the formula of PSO to perform this strategy.

$$V_j' = wV_j + c_{1j}r_1\left(p_j - x_j\right) + c_{2j}r_2\left(p_n - x_n\right) \tag{6}$$

Vj denotes the velocity which measures the individual j's position changing. The *w* denotes the position weight; it represents the degree of active personality. The position weight is *characteristic* that we mention in the Initial Individual phase. When the weight is high, it means the individual want to interact and communicate with other individuals, so the individual would like to move violently on the plane and try to find comfortable position.

c_{1j} and c_{2j} are the degree of the individual j's decision, which affected by personal or the group's experience separately. *r1* and *r2* are random numbers between zero and one that mentioned in the original formula in PSO. *Vmax* denotes the upper-bound of the velocity of the individuals' movement in PSO and the scope of individual's energy in certain iteration.(Fig. 3)

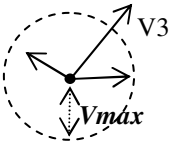


Fig. 3 This shows the scope of upper-bound energy where individual j can reach within the dotted circle. *V3* will be modified to *Vmax* because it is outside the scope of j's energy

The position of individual j will be updated as following.

$$x_j' = x_j + V_j' \tag{7}$$

2. Energy-changing strategy:

Changing the radius will affect the number of connections, and cause the change of cost. To make the individual more comfortable, the individual can reduce the radius to cut out the links or increase the radius to establish more links. By means of the parameter characteristic, it denotes the individual's willingness to establish connection with other individuals.

4 Moving Path Simulation

With the proposed framework, we simulate several connectivity distributions on the plane and show the result of steps generated by PSO in the following sections.

4.1 Evolution Process

We simulate particle swarm by using the processes in section3. We can see the individuals scattering on the internet with his own energy. There's a link been established when any of the individuals can afford the cost of building connection.

After executing 10 runs of the simulation, we randomly show the evolution process of 9 iterations in Fig.4. Obviously, the particle paths converge together from (a) to (c), the superficial measure of the cycle becomes bigger. In the "initialization" step, the number of links is fewer, which results in the low cost, and individuals are in the comfortable state more easily. We can tell that the individual's characteristic is full of passion, which could increase the individual's energy to establish connections with others. After "Self-Adaption", there're more chances that the energy of individuals is higher and get along with other individuals.

When it comes to (d), (e) and (f), some individuals still had the trend of convergence, but some individuals started to diversify. In order to feel more comfortable, the individuals start to diminish the energy. And the individuals are less eager to establish the link than usual.

From (g) to (i), the trend of diversity is more significant. But there're some individuals who are full of passion and eager to establish the links with other individuals. We consider that they are the few numbers of contributors in the internet.

Meanwhile, there's another phenomenon been found. Although the individuals diversify and the energy is smaller with time, most individuals converge to some blocks and its shape of distribution is scatter on the left side in Fig.4.

4.2 Overview Result

Every individual owns his free will, the interaction between two individuals or the individual and the environment is an unique model. Even if single individual has

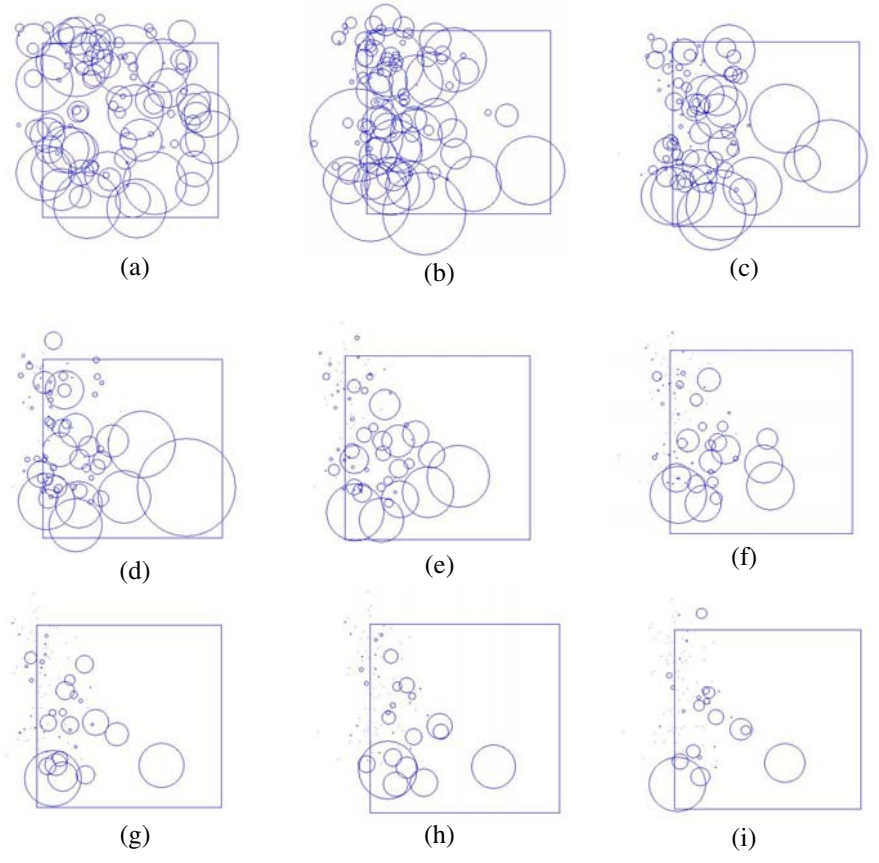


Fig. 4 The results of 9 iterations

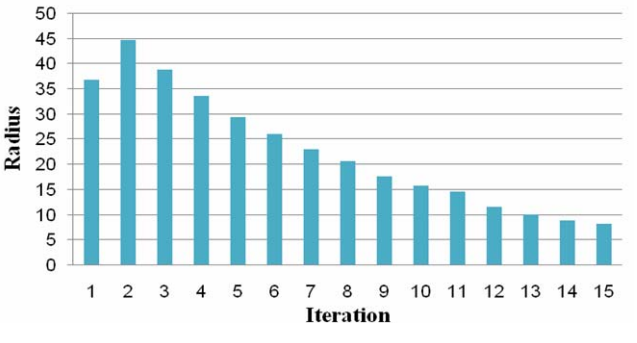


Fig. 5 Average radius of 100 individuals' observation in 30 runs for the group's behavior

his own behavior model as mentioned before, but when it comes to the whole group, the behavior model of group will follow some patterns. The individuals' average energy will be faded with the passing of time; it means that most individuals' radius, i.e., the passion to interact with other individual, might be smaller. Meanwhile, we can sum up a regular pattern of group behavior.

5 Conclusion

In our research, we observe the result that individuals are affected by the human society decision. The behavior of the individuals in the virtual community doesn't totally conform to the expectation of companies in the Web 2.0 environment. In the long-term observation, there are merely individuals lasting their energy for a long period of time. Most individuals will be tired of being engaged in social activities, and their energy will be lower. Finally, they will only communicate with the neighbors.

But in the strategy of web 2.0 companies, they usually ignore this phenomenon. The decision maker should think about other important factors that influence the virtual community. For example, web 2.0 company might need to think about how to provide a prefect communication platform such that users can communicate more easily when providing service to ensure the convenience of communications and connections if needed. Enriching the information content or providing unique service, which needs higher transition cost, might be the direction of strategies.

References

[1] Barnes, J.A.: Class and Committees in a Norwegian Island Parish. Human Relations 7(1), 39–58 (1954)
[2] George, A.A.: Social Distance and Social Decisions. Econometrica 65(5), 1005–1027 (1997)
[3] Granovetter, M.: The Strength of Weak Ties: A Network Theory Revisited. In: Marsden, P., Lin, N. (eds.) Social Structure and Network Analysis, p. 105. Beverly Hills (1982)
[4] Guare, J.: Six Degrees of Separation: A Play. Vintage Books, New York (1990)
[5] Milgram, S.: The small-world problem. Psychology Today 1, 60–67 (1967)
[6] Miller, P.: Web 2.0: Building the New Library. Ariadne (45) (2005), http://www.ariadne.ac.uk/issue45/miller/
[7] O'Reilly, T.: What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. O'Reilly, Sebastopol (2005), http://www.oreillynet.com/lpt/a/6228
[8] Rheingold, H.: The Virtual Community (1993)
[9] Thomas, C.S.: The Strategy of Conflict. Harvard University Press, Cambridge (1960)
[10] Davis, I., Talis.: Web 2.0 and all that (2005), http://iandavis.com/blog/2005/07/talis-web-20-and-all-that?year=2005&monthnum=07&name=talis-web-20-and-all-that
[11] Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)

[12] Tversky, A., Kahneman, D.: Availability: A heuristic for judging frequency and probability. Cognitive Psychology 5, 207–232 (1973)
[13] Wellman, B., Hampton, K.: Living Networked On and Offline. Contemporary Sociology 28(6), 648–654 (1999)
[14] Amaral, L.A.N., Scala, A., Bathélémy, M., Stanley, H.E.: Classes of behavior of small-world networks (2000), http://xxx.lanl.gov/cond-mat/0001458

Genetic Algorithms in Chemistry: Success or Failure Is in the Genes

Clifford W. Padgett and Ashraf Saad

Abstract. In many areas of chemistry there are problems to which genetic algorithms (GA's) can easily be applied. Several chemistry problems in which GA's are used will be examined. Currently they are used mainly for the generation of regression curves, protein folding, structure elucidation, parameterizations, and system optimization. Perhaps it is the GA simplicity and ease of use that has facilitated the widespread use of the soft computing method in chemistry. This paper focuses on how GA's have been modified to solve discipline specific problems in the chemical sciences.

1 General Introduction to Genetic Algorithms in Chemistry

In the early 1960's, biologists began experimenting with computer simulations of genetic systems. Holland is considered the father of this subfield of computer science for his early work with genetic algorithms [1]. However, it was not until the 1980's that GA's received significant attention in the applied sciences. In the late 1980's, GA's were brought into the field of chemometrics by Lucasius and Kateman [2]. This opened the door for the application of GA's to problems in chemistry.

In many areas of chemistry there are problems to which genetic algorithms (GA's) can be readily applied. For instance, one primary focus in the subfield of analytical chemistry is the determination of the identity and amount of a given compound in an unknown mixture via spectroscopic methods that probe the molecules using electromagnetic radiation. Most of these methods generate spectra that contain information about the molecule but it is not always easy to extract the required data from the spectra. Genetic algorithms have been successfully applied here as they are efficient at finding patterns in data even when the data contains a large amount of extraneous information. GA's have also been used for the generation of regression curves, protein folding, structure elucidation, parameterizations, and for system optimization. Since then, the number of uses for GA's in chemistry

Clifford W. Padgett
Department of Chemistry and Physics Armstrong Atlantic State University Savannah, GA 31419, USA

Ashraf Saad
Department of Computer Science Armstrong Atlantic State University Savannah, GA 31419, USA

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 181 – 189.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

has steadily increased, especially as a calibration technique and as a wavelength selection method, such as component concentration determination from IR spectra [3]. In the late 1990's, GA's made their way into the field of crystallography as a tool for structure elucidation [4, 5].

2 Types of Problems in Chemistry Where Genetic Algorithms Are Used

The use of GA's in chemistry falls into three main categories, regression, configuration methods, and data mining. Regression methods use experimental data (such as an infrared spectrum of a compound) to determine the parameters of an equation that relates to a quantitative property like concentration. Configuration methods use GA's to manipulate the geometry of a molecule to fit experimental data or minimize energy. Some examples include protein folding and structure elucidation. Data mining methods use GA's to determine information that can be used to find relationships or parameterize other methods.

2.1 Genetic Regression

Genetic Regression (GR) is the application of the genetic algorithm to the problem of generating a calibration curve. This method, which has been described in detail in several papers [8,9], is based on the optimization of linear regression with the use of a genetic algorithm. The GA combines data points from spectra in a manner that allows for a simple linear regression curve to be generated. GR has been shown to be effective in finding good calibration models for data with noise, baseline fluctuations, overlapping spectral features [8,9], and spectral drift as well as in multi-instrument calibration [10]. GR has also been shown to be effective at searching through large solution spaces and eliminating over 99.9999% of the unsuitable solutions with little computational requirements [11]. It has been compared and combined with more familiar multivariate methods such as PLS, ILS [12] and has been shown to be comparable to or better than multivariable methods [7].

As with most GA's, the GA used in a typical GR follows the same basic steps, and GR implements a GA that optimizes simple linear regression models through an evolving selection of wavelengths. The GR program created by Paradkar and Williams [8] has four major control parameters: r-squared value, mutation rate, number of iterations, and number of genes used. The r-squared value is used to screen the initial genes by forcing them to be created with minimum correlation, thus the randomly initialized genes in the gene pool must have a fitness value below some minimal value. The mutation rate determines how often the genes are mutated during the mutation step and the rate goes from 0% (no mutation) to 100% (mutates at every mutation step). The number of iterations sets how many times the program repeats its steps, and the number of genes determines how many genes are evaluated during the run.

First, the algorithm initializes a population of genes, with each gene being two to five bases long. The bases of the genes are composed of two wavelengths and a mathematical operator to combine them. The GA randomly selects the number of bases for each gene and the operator and two wavelengths used in each base.

Several operators are available (addition, subtraction, multiplication, division) and most spectra have up to ten thousand wavelength points from which to select. A typical example would be as follows:

$$S = (I_{7202} - I_{6457}) + (I_{7690} - I_{7110})$$

Where S is the so-called genetic score of the gene and I is the instrument response (absorbance, reflectance, etc.) at the wavelength in the subscript, Figure 1 shows this gene pictorially. Next the algorithm evaluates the population (typical size is around 50 genes) by calculating the standard error of calibration (SEC) for each gene.

$$SEC = \sqrt{\frac{\sum(\overline{Y} - Y)^2}{n - 2}} \quad (1)$$

Where $\overline{Y}$ is the calculated value, Y is the true value, and n is the number of samples in the calibration set. The SEC is used as the measure of fitness of the genes.

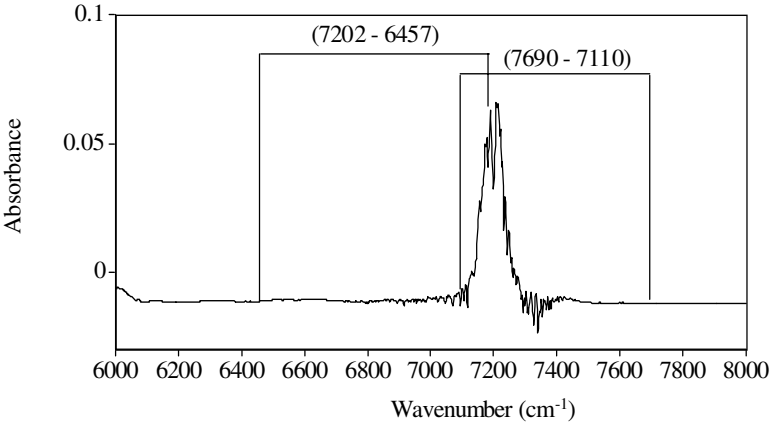


Fig. 1 Illustration of a gene on IR absorption spectrum

The scores are generated for all spectra in the calibration set, and then regressed against a known concentration to generate a calibration curve by using a simple least squares method to build the model. Then the previously calculated SEC's are used to rank the genes based on their fitness. They are ranked from lowest (best) SEC to highest (worst) SEC. Genes are then selected for breeding based on their fitness. A roulette wheel selection method is used where the chance of selection is directly proportional to the gene's fitness. Each gene can be selected multiple times; however, genes with higher fitness have a better chance of being selected than genes with lower fitness values. They are allowed to mate top-down after gene selection: the first gene (G_1) is mated with second gene (G_2), third gene is mated with fourth gene, and so on until all are mated. In the mating step, the paired genes are broken at random points followed by single point crossover. This

creates two new genes from the information contained in the old genes. This process can be illustrated as follows:

$$G_1 = (R_{2564} + R_{824}) + (R_{2190} * R_{1735}) \# + (R_{1075} / R_{2004}) + (R_{2964} / R_{1185})$$
$$G_2 = (R_{1034} / R_{824}) \# + (R_{2533} * R_{1512}) + (R_{1846} / R_{2234})$$

Where G_1 and G_2 are the two parent genes, and the points where the genes are cut for mating/single-point crossover are marked by #. The first piece of G_1 is combined with the second piece of G_2. Likewise, the first piece of G_2 and the second piece of G_1 are combined. This yields two new daughter genes that are:

$$G_3 = (R_{2564} + R_{824}) + (R_{2190} * R_{1735}) + (R_{2533} * R_{1512}) + (R_{1846} / R_{2234})$$
$$G_4 = (R_{1034} / R_{824}) + (R_{1075} / R_{2004}) + (R_{2964} / R_{1185})$$

Mating can increase or decrease the number of base pairs in the daughter genes. The GR program was designed to maintain a minimum of two base pairs in a gene so that breeding would always be possible and a maximum of five base pairs was set during initialization to speed up the initial selection.

After breeding the genes are allowed to mutate at a user-specified rate. When a gene mutates, one of its bases is randomly replaced with a new randomly-initialized base pair. This process adds diversity to the gene pool and helps prevent the algorithm from becoming trapped in a local minimum. After the new genes are created and evaluated, the old genes are replaced with a combination of the fittest new ones and the fittest old ones (elitism). These processes then repeat until the termination condition is met, i.e., when the algorithm has completed a fixed number of iterations.

2.2 Structure Elucidation

Genetic algorithms are also being used to generate possible 3-dimensional arrangements of atoms in space and checking them against X-ray diffraction data. Typically, X-ray powder diffraction data is used as the experimental data to check the fitness of the genes. Determination of crystal structures from single crystal X-ray diffraction data is one of the most important tools for elucidating structural information in chemistry. X-ray single crystal methods give the clearest picture of both molecular structure and solid-state structure of an analyte. However, single crystal X-ray diffraction has limitations imposed by its very nature; many materials of interest cannot be prepared as single crystals of sufficient size and quality. It is possible to generate microcrystalline powders of most materials and although they are not amenable to single crystal methods, they can be examined via X-ray powder techniques. Methods for structure elucidation from powder diffraction data have had some success with the application of global optimization methods such as Monte Carlo [13], simulated annealing [14], or genetic algorithms [15]. These methods are called the direct-space methods. They get their name from the fact that they search the direct space; i.e., real space as opposed to reciprocal space that is typically used in traditional X-ray methods. Unfortunately, there are major difficulties associated with determining crystal structures directly from the powder

diffraction data, primarily originating from peak overlap in, and preferred orientation of, the data. The peak overlap problem stems from the tendency of organic compounds to crystallize with large unit cells of low symmetry, resulting in many closely-spaced diffraction images, and from the compression of the three-dimensional information into a one-dimensional powder pattern. Difficulties with preferred orientation are common to all compounds and are a result of the shape of the crystals in the microcrystalline powder. Direct-space methods have been proposed and shown [13, 14, 15, 16 ,17] to provide a rational basis for overcoming difficulties in structural elucidation from powder patterns.

In a typical direct-space method, the program begins with a rigid molecule, or one with only a few internal degrees of freedom, and the unit cell parameters, which are obtained by indexing the powder pattern. Next, the optimization method chosen inserts this molecule into the unit cell and calculates the powder pattern of the trial solution using a powder prediction-algorithm like Powd12 [18] The calculated X-ray powder pattern is compared to the experimental one and a figure of merit is calculated for the trial structure. The preferred figure of merit is Rwp [19] (residual weighted error) and is defined as:

$$Rwp = 100 \times \sqrt{\frac{\sum_i w_i (Y_{Exp_i} - Y_{Cal_i})^2}{\sum_i w_i Y_{Exp_i}^2}} \tag{2}$$

Where w_i is a weighting factor related to the experimental error at point i, Y_{Expi} is the ith point from the experimental spectrum, and Y_{Cali} is the ith point from the calculated structure. Other figures of merit include packing energy considerations in addition to Rwp in order to eliminate implausible structures. Once a structure has been determined that closely matches the experimental data, Rietveld refinement of the model is carried out to convergence [19]. The resulting solution can be comparable to that obtained by single crystal methods.

An example of one such implementation of a GA is OCEANA [20]. OCEANA performs a grid search over unit cell parameters and a user-defined space-group and uses a genetic algorithm to search for an optimal fit to the experimental X-ray powder pattern. The GA generates genes similar to the one shown in figure 2, which encode the information necessary to describe a crystal structure of the compound. The GA determines the fitness of the genes using Rwp (above) and trial structure energies (calculated using the atom-atom potential method).

The GA in OCEANA first generates a random population of genes with a fixed size (user-defined). However, the GA also has means to preferentially select genes with more information in them. The Rwp value of the genes is one such way it does this; genes that have too high of an Rwp value are removed and more are generated until the preset number of genes with Rwp values below the cutoff value has been reached. The packing energy is another way to limit the initial population of genes. All genes must have an energetically favorable (user-defined) packing energy.

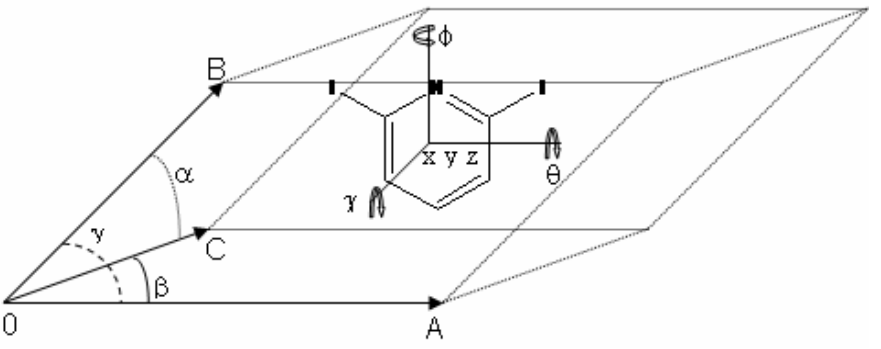


Fig. 2 Illustration of a gene. G = {A, B, C, α, β, γ, x, y, z, χ, θ, ϕ}

The genes of the OCEANA program consist of twelve base pairs. Six bases represent the cell parameters and six represent the position and orientation of the molecule in the unit cell. More bases can be added to handle internal degrees of freedom for the molecule and for additional molecules in the asymmetric unit. These bases are the elemental unit of the OCEANA program, as they are the objects that are randomly mutated and exchanged during breeding, (the A B C … in the GA example above).

In the OCEANA program, genes are ranked by their Rwp and their energy. The Rwp is calculated by the equation above, and the packing energy is calculated by a Buckingham potential modified to include charge using the atom-atom potential method [21]. The (Exp-6-1) Buckingham equation is shown below:

$$E_{ij} = Ae^{-Br_{ij}} - \frac{C}{r^6} + \frac{q_i q_j}{r} \tag{3}$$

Where A, B, C are constants for each atom type that are fitted to the experiment, and the q's are calculated by fitting to electrostatic potential.

Once the genes have been created, they are ranked by a ninety percent-ten percent combination of Rwp and Energy. First the packing energy and Rwp are normalized to scale them between 0 and 1. Genes which produce a low Rwp and a low energy have a higher rank than those that do not. After the initial genes have been ranked, they undergo natural selection, using tournament or roulette wheel selection, with single-point crossover breeding or multipoint crossover (selection and breeding modes are user-defined) followed by two mutations steps. The mutation rates for these two different mutation steps are set before running the program. The first mutation rate determines how often a base is replaced with a randomly generated one. The second mutation rate determines how often a base is replaced with a randomly generated constrained one (randomly generated but required to be near the original value). Note that OCEANA uses elitism, so even a 100% mutation rate will not destroy good genes. Genes are optimized using a simplex minimizer to invoke Lamarckian evolution.

After the preset number of iterations, the OCEANA program stops and uses the best gene to generate a trial structure model. This is done by using the first six bases as the unit cell and the next six to place the molecule in the unit cell. This

trial structure is then saved for possible further refinement, and as a representation of the current grid point.

2.3 Other Uses

GA's have found uses in most areas of chemistry and have been a very successful computing tool for chemistry. Another area that has been and still is being majorly influenced by GA's is drug design in which combinatorial chemistry is now being done with a GA to optimize the search for the best drug for the receptor [22]. GA's are also having an impact on many fields through the use of GA in parameter optimizations, for example GA's have been used to parameterize semi-empirical methods [23] that were then used in a molecular dynamic simulation. GA's have had widespread use in the determination of the three-dimensional structures of proteins from their amino acid sequences [24, 25, 26, 27]. This problem is perhaps one of the most difficult in chemistry to which GA's have been applied. The uses of GA's in chemistry are too numerous to list all of them. It is hoped that the ones listed here give the reader an idea how diverse their uses are and that they are becoming a mainstream technique in many areas of chemistry.

3 Conclusions

GA's have been so successful at solving many problems in chemistry that there is no doubt that they will remain a mainstay in many areas. There are many general, simple to use GA codes available online for chemists to use; additional GA's are simple to construct and the authors have had great success at getting undergraduate students to write and modify them for specific problems. Unfortunately, GA's are not the solution to every problem and have many pitfalls. Perhaps the biggest problems with GA's are premature convergence and over-fitting. Premature convergence is where a population converges on a local minimum that is not an ideal solution to the problem at hand. Over-fitting is where the GA finds an excellent fit to a set of calibration data, but, when applied to data not in the calibration set, the genes perform horribly. While premature convergence can generally be surmounted by optimizing the GA's parameters for the specific problem, over-fitting requires more drastic measures to circumvent. Generally, the use of a validation set is required, where some data is not used by the GA to generate genes but the extra data is used every now and then to evaluate gene performance. Even with the limitations, GA's are having a great impact in chemistry and many people are constantly coming up with new ways to use GA's and to circumvent the current problems associated with their use. The future of GA's in chemistry will most likely be the use of hybrid soft computing methods, an indeed such methods are already being employed [28,29,30]. The use of such hybrid methods will continue to expand the usefulness of GA's in chemistry for years to come.

References

1. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
2. Lucasius, C.B., Kateman, G.: Genetic algorithms for large-scale optimization in chemometrics: An application. Trends Anal. Chem. 10, 254–261 (1991)
3. Ozdemir, D., Mosley, M., Willams, R.: Hybrid Calibration Models An Alternative to Calibration. Transfer. Appl. Spec. 52, 599–603 (1998)
4. Harris, K.D.M., Kariuki, B.M., Tremayne, M., Johnston, R.L.: New methodologies for solving crystal structures from powder diffraction data. Mol. Cryst. Liq. Cryst. 313, 1–14 (1998)
5. Bazterra, V.E., Ferraro, M.B., Facelli, J.C.: Modified genetic algorithm to model crystal structures. I. Benzene, naphthalene and anthracene. J. Chem. Phys. 116, 5984–5991 (2002)
6. Lawerence, D. (ed.): Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York (1991)
7. Bauer Jr., R.J. (ed.): Genetic Algorithms and Investment Strategies. John Wiley and Sons, Inc, New York (1994)
8. Paradkar, R.P., Williams, R.R.: Genetic Regression as a Calibration Technique for Solid-Phase Extraction of Dithizone–Metal Chelates. Appl. Spectrosc. 50, 753–758 (1996)
9. Paradkar, R.P., Williams, R.R.: Correcting Fluctuating Baselines and Spectral Overlap with Genetic Regression. Appl. Spectrosc. 51, 92–100 (1997)
10. Ozdemir, D., Mosley, M., Williams, R.R.: Effect of Wavelength Drift on Single- and Multi-Instrument Calibration Using Genetic Regression. Appl. Spectrosc. 52, 1203–1209 (1998)
11. Mosley, M., Williams, R.R.: Determination of the Accuracy and Efficiency of Genetic Regression. Appl. Spectrosc. 52, 1197–1202 (1998)
12. Jouan-Rimbaud, D., Massart, D., Leardi, R., De Noord, O.E.: Genetic Algorithms as a Tool for Wavelength Selection in Multivariate Calibration. Anal. Chem. 67, 4295–4301 (1995)
13. Harris, K.D.M., Tremayne, M., Lightfoot, P., Bruce, P.G.: Crystal Structure Determination from Powder Diffraction Data by Monte Carlo Methods. J. Am. Chem. Soc. 116, 3543–3547 (1994)
14. Engel, G.E., Wilke, S., König, O., Harris, K.M.D., Leusen, F.J.J.: PowderSolve - a complete package for crystal structure solution from powder diffraction patterns. J. Appl. Crystal-logr. 32, 1169–1179 (1999)
15. Harris, K.D.M., Johnston, R.L., Kariuki, B.K.: The Genetic Algorithm: Foundations and Applications in Structure Solution from Powder Diffraction Data. Acta Crystallogr. A54, 632–645 (1998)
16. David, W.I.F., Shankland, K., Shankland, N.: Routine determination of molecular crystal structures from powder diffraction data. Chem. Commun. 8, 931–932 (1998)
17. Shankland, K., David, W.I.F., Csoka, T., McBride, L.: Structure solution of Ibuprofen from powder diffraction data by the application of a genetic algorithm combined with prior con-formational analysis. Int. J. Pharm. 165, 117–126 (1998)
18. Smith, D.K., Nichols, M.C., Zolensky, M.E.: POWD12, A FORTRAN IV Program for Calculating X-ray Powder Diffraction Patterns. Version 12. The Pennsylvania State University: University Park, PA (1982)

19. Rietveld, H.M.: A Profile Refinement Method for Nuclear and Magnetic Structures. J. Appl. Cryst. 2, 65–71 (1969)
20. Padgett, C.W., Arman, H.D., Pennington, W.T.: Crystal Structure Elucidated from X-ray Powder Diffraction Data without Prior Indexing. Crystal Growth & Design 7, 367–372 (2007)
21. Pertsin, A.J., Kitaigorodsky, K.I.: The Atom-Atom Potential Method. Applications to Organic Molecular Solids. Springer, Heidelberg (1987)
22. Weber, L.: Evolutionary combinatorial chemistry: application of genetic algorithms. Drug Discovery Today 3, 379–385 (1998)
23. Sastry, K., Johnson, D.D., Thompson, A.L., Goldberg, D.E., Martinez, T.J., Leiding, J., Owens, J.: Multibojective Genetic Algorithms for Multiscaling Excited State Direct Dynamics in Photochemistry. In: Proceedings of the Genetic and Evolutionary Computation Conference (2006)
24. Sun, S.: Reduced representation model of protein structure prediction: Statistical potential and genetic algorithms. Protein Sci. 2, 762–785 (1993)
25. Sun, S.: Reduced representation approach to protein tertiary structure prediction: Statistical potential and simulated annealing. J. Theor. Biol. 172, 13–32 (1995)
26. Kolinski, A., Skolnick, J.: Monte Carlo simulations of protein folding. I. Lattice model and interaction scheme. Proteins 18, 338–352 (1994)
27. Rey, A., Skolnick, J.: Computer simulations of the folding of coiled coils. J. Chem. Phys. 100, 2267–2276 (1994)
28. Zhao, X.: Advances on protein folding simulations based on the lattice HP models with natural computing. Applied Soft Computing 8, 1029–1040 (2008)
29. Oduguwa, A., Tiwari, A., Roy, R., Bessant, C.: An Overview of Soft Computing Techniques Used in the Drug Discovery Process. Applied Soft Computing Technologies: The Challenge of Complexity, 465–480 (2006)
30. Abraham, A., Grosan, C.: Soft Computing for Modeling and Simulation. J. Simulation Systems, Science and Technology 6, 1–3 (2005)

Multi-objective Expansion Planning of Electrical Distribution Networks Using Comprehensive Learning Particle Swarm Optimization

Sanjib Ganguly, N.C. Sahoo*, and D. Das

Abstract. In this paper, a Pareto-based multi-objective optimization algorithm using Comprehensive Learning Particle Swarm Optimization (CLPSO) is proposed for expansion planning of electrical distribution networks. The two conflicting objectives are: installation and operational cost, and fault/failure cost. A novel cost-biased particle encoding/decoding scheme, along with heuristics-based conductor size selection, for CLPSO is proposed to obtain optimum network topology. Simultaneous optimization of network topology, reserve-branch installation and conductor sizes are the key features of the proposed algorithm. A set of non-dominated solutions, capable of providing the utility with enough design choices, can be obtained by this planning algorithm. Results on a practical power system are presented along with statistical hypothesis tests to validate the proposed algorithm.

1 Introduction

Computerized distribution system expansion planning is of active research interests among distribution engineers owing to two major benefits, i.e., minimization of expansion cost and reduction of system's fault/failure. The planning aims to optimize network design parameters, i.e., network topology, branch conductor sizes, and number and location of reserve-branches so as to enhance network reliability. A number of approaches to this planning problem has been proposed in the past. In [1-3], only cost is optimized, while various reliability aspects are considered in [4-9]. Cost and reliability conflict with each other. Thus, multi-objective optimization by considering these two as separate objectives with a suitable trade-off analysis

Sanjib Ganguly · N.C. Sahoo · D. Das
Department of Electrical Engineering,
Indian Institute of Technology,
Kharagpur-721302, India
e-mail: ncsahoo@ee.iitkgp.ernet.in

* Corresponding author.

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 193–202.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

is the most practical approach. In [8-9], the Pareto-optimal strategy has been used to trade off these conflicting objectives. In both the works, genetic algorithm with direct encoding scheme (of chromosome) is used; but no performance assessment of the stochastic optimizer is shown.

In this work, a different stochastic optimizer, i.e., Comprehensive Learning Particle Swarm Optimization (CLPSO) is used. The CLPSO is an advanced version of particle swarm optimization (PSO) proposed by Liang et. al [12]. PSO [10] is a well-known class of evolutionary algorithm with superior convergence and applied in several multi-objective problems. A state-of-the-art review on multi-objective PSO (MOPSO) based on Pareto-optimality principle can be found in [13]. The main goal is to reach towards well diversified true Pareto-optimal solutions. Several types of PSO topologies (ring, fully connected) are proposed to improve the swarm performance. To promote diversity, various strategies such as, preserving elite solutions in different ways, usage of mutation operator and nearest neighbor density estimator, clustering of whole swarm in several sub-swarms, and NSGA II based fitness assignment are used.

In this work, Strength Pareto Evolutionary Algorithm (SPEA) and its improved version SPEA2 [18] are applied to assign particle fitness and promote diversity. A novel indirect scheme, i.e., cost-biased particle encoding/decoding for realization of networks, is proposed to reduce the problem dimension considerably. The branch conductor sizes are optimized simultaneously along with network topology. The results are compared with those of the GA based planning [9]. Further, a statistical hypothesis test is performed to assess the stochastic performance of the algorithm.

The organization of paper is as follows. Section 2 deals with problem formulation. CLPSO with proposed particle encoding is discussed in Section 3. Sections 4 and 5 provide details of multi-objective optimization and overall planning algorithm, respectively. Section 6 presents simulation results and performance assessment. Section 7 concludes the paper.

2 Problem Formulation

Distribution system expansion planning can be categorized as: (i) single-stage planning and (ii) multi-stage planning. The former is a one-time optimization technique whereas, in the latter, a single-stage optimization is performed at each stage for the overall network optimization. A single-stage expansion planning model proposed here consists of two objective functions, i.e., (i) installation and operational cost, and (ii) failure cost. The latter is an aggregate of costs due to feeder failure/outage and non-delivered energy (NDE). The former is an aggregate of:

* Installation cost of new substations
* Cost of incremental capacity addition of existing substations (if required)
* Installation cost of new feeders or additional branches in existing feeders
* Cost of conductor replacement of existing branches (if required)
* Annual maintenance cost of feeders
* Cost of feeder energy losses

The two objective functions are formalized as:

$$C_{Fi} = \sum_{j=1}^{N_b} C^{I_b} l_j + \sum_{j=1}^{E_b} C^R l_j + \sum_{j=1}^{(N_b+E_b)} (C^{M_f} l_j + C^V) t_a + \sum_{k=1}^{N_s} C_k^{I_s} + \sum_{k=1}^{E_s} C_k^{IC} \quad (1)$$

$$C_{Fa} = \sum_{j=1}^{N_b} (C^F l_j \lambda_j + C^{NDE} d_j t_a) P_j L_i + \sum_{j=1}^{E_b} (C^F l_j \lambda_j + C^{NDE} d_j t_a) P_j L_i \quad (2)$$

where, C_{Fi}: Total installation and operational cost ; C_{Fa}: Failure/fault cost
N_b and E_b: Number of additional and existing branches, respectively
N_s and E_s: Number of additional and existing substations respectively
C^{I_b} (C^{M_b}): Branch installation (annual maintenance) cost per unit length
C^V: Cost of energy losses; t_a: Analysis time
C^R: Conductor replacement cost of branch per unit length l_j: Length of branch j
C^{IC}: Substation incremental capacity addition cost
C^F and C^{NDE}: Outage cost and cost of non-delivered energy, respectively
λ_j and d_j: Feeder failure rate and duration at branch j, respectively
P_j: Power flow through branch j; L_i: Importance level of node i
$\forall(N_b, E_b)$ and (N_s, E_s) belong to a set of predefined sizes

The planning problem is to minimize these two objective functions subject to following constraints:

(i) Power balance constraint (i.e., demand and supply balance)
(ii) Substation capacity constraint
(iii) Feeder capacity constraint,
(iv) Node voltage deviation constraint
(v) Radiality constraint: designed network should be radial.

3 PSO, CLPSO and Particle Encoding/Decoding

PSO is a population based multi-point search technique inspired by social behavior of bird flock [10]. Like other evolutionary algorithms, the search starts with a population of search points, called particles. Each particle is encoded as a position vector containing n dimensional information (initially chosen randomly) and the position of each dimension is updated with velocity (initially chosen randomly) information in successive iterations. The velocity vector of the particle is updated by following its own previous best value called *pbest* and by learning the best particle of population called *gbest*. The position of a particle is updated in each iteration as:

$$V_{id} = \omega V_{id} + \phi_1 r_1 (pbest_{id} - X_{id}) + \phi_2 r_2 (gbest_{id} - X_{id}); \;\; X_{id} = X_{id} + V_{id} \quad (3)$$

[i=1,2,...,Population Size, and d=1,2,...,Maximum Dimension]
where, X_{id} and V_{id}: position and velocity of i-th particle's d-th dimension
ϕ_1 and ϕ_2 : learning constants; r_1 and r_2 : random numbers in interval [0,1]
pbest and *gbest*: Particle's own best and group best, respectively
ω : a linearly decreasing inertia weight

The two learning constants influence the advancements of particles towards the *pbest* and *gbest*. Inertia weight is used to control the impact of previous velocity to provide a balance between local and global search. In PSO, particle gets very often trapped in local optima as every particle follows the group leader.

3.1 CLPSO [12]

CLPSO, proposed by Liang et. al., is essentially a local search technique, having similar computational complexity as PSO. It is not influenced by *gbest*, therefore less chance of being trapped into local optima. For each dimension, a particle updates its position by either following its own *pbest* or by following different particles, reasonably fitter in the corresponding dimensions (i.e., having better overall fitness). The strategy is to update all particle's individual dimensions separately by learning from different exemplars. The pseudocodes for updating a particle are:

```
 d=1
While d<=Maximum_Dimension
 if rand<LP_{ci}%rand=random no.,LP_{ci}= learning probability
  Update position and velocity using Eq.(1.3) with phi_{2}=0
 else
  Choose two particle (p & q) randomly, compare their fitness
  Use the winners pbest as exemplar to learn for the dimension
  {update position and velocity by Eq.(1.3) with pbest_{id}
  replaced by pbest_{pd} and phi_{2} = 0)}
 endif
 d=d+1
endwhile
```

The motivation behind using CLPSO is its superiority over other PSO variants for multimodal problems, as illustrated in [12]. In [12], it is shown that, in various multimodal problems, this learning strategy explores larger potential search space than PSO. This is because *gbest* in PSO always influences the particles to move towards global best. Hence, better diversity should be achievable in CLPSO.

3.2 Proposed Encoding Scheme

There are two types of particle encoding schemes used in PSO for combinatorial optimization problems, i.e., direct and indirect encoding. The former directly represents solution, while the latter only carries some guiding information to obtain solution rather than the solution itself.

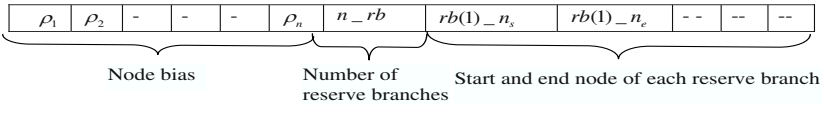


Fig. 1 Proposed particle encoding scheme

In this work, the particle position vector is designed to consist of two distinct segments (Fig. 1): (i) network segment consisting of node bias values ρ [$\rho \in$(-1.0, 1.0)], (ii) reserve-branch segment consisting of two sub-segments, i.e., number of reserve-branches, start and end nodes of each reserve-branch. The reserve branches enhance reliability of the network. The maximum number of reserve-branches is problem-specific and initial choice of the nodes of the reserve-branches are random and then are updated (in integer mode) in CLPSO.

3.3 Proposed Decoding Scheme

The decoding scheme used here is based on a strategy proposed in [11], called cost-biased decoding, where nodes are sequentially selected and appended to the terminal node of the growing path on the basis of minimum value of the product of branch costs and node bias values: $j = argmin(C_{ij}\rho_j)$ [C_{ij}: cost of branch $i - j$ and ρ_j is the bias value of node j connected to node i]. Inspired by this strategy, a novel cost-biased decoding scheme is proposed to generate radial network with main/forward and lateral branches. The pseudocodes of proposed decoding scheme are as follows:

```
Let {Q}: Nodes of branches existing in the network,
{R}: Nodes to be connected with the existing network
{Ns},{Ne}:Start and end nodes of existing branches,respectively
 while (R != null)
     for i=1,...,length(Q)
        for j=1,...,length(R)
         C(i,j)=D(Q(i),R(j))*p(j) [D:Distance and p:bias]
        endfor
     endfor
  Find the minimum element of C and corresponding Q(i) and R(j)
  Update {Ns}<-{Ns,Q(i)};  Update {Ne}<-{Ne,R(j)}
  Update {Q}<-{Q,R(j)} ; delete R(j) from {R}
endwhile
```

The decoding of the reserve-branch segment is, however, simple due to direct encoding of number of such branches with their end nodes. Some heuristics are used to filter out the invalid connections such as branch of main network appearing as a reserve branch, same start and end node etc. It is noteworthy that same network topology may repeat in several iterations as many particles may map to same network; but they might have different sets of conductors which are selected randomly (see section 5). Thus, their fitness might be different.

4 Multi-objective Optimization

In multi-optimization, goal is to optimize all the objectives simultaneously. In this problem, two objectives, i.e., total installation (and operational) cost, and total failure cost, conflict with each other. In multi-objective problem with conflicting objectives, no solution can improve itself in one objective without worsening in at least

one other objective. Therefore, a suitable trade-off of objectives is required and this is usually done by Pareto-optimality principle.

4.1 Pareto-Optimality Principle

The Pareto-optimality principle states that, for an m objective minimization problem, a solution x is said to dominate solution y ($x \prec y$) iff:

$$\forall i \; f_i(x) \leq f_i(y) \text{ and } \exists j \; f_j(x) < f_j(y) \; [i = 1, 2, ...m]$$

All the solutions which are not dominated by any other solution, are called non-dominated solutions/elite solutions and are usually stored in Elite Archive.

4.2 Fitness Assignment and Elite Preserving Scheme

The performance of the multi-objective stochastic algorithm depends on: (i) fitness assignment scheme to guide all population members towards Pareto optimal solutions, and (ii) elite preserving scheme to diversify the search space. In this work, an efficient algorithm, i.e., Strength Pareto Evolutionary Algorithm (SPEA) of Zitzler and Thiele and its improved version SPEA2 [18] are followed (separately) to fulfill this dual purpose. The comparative main features of these algorithms are:

Table 1 Comparison between SPEA and SPEA2

Algorithm	Fitness assignment	Elite preserving scheme
SPEA	An elite member's strength is proportional to number of current (population) members dominated by it. A current member's fitness is the sum of its dominators strengths.	Clustering Technique
SPEA2	Strength is assigned to all members (current and elite). Fitness of each member of current population is sum of its dominator's strength plus density around it.	Archive Truncation

5 Complete Distribution System Expansion Planning Algorithm

The complete planning algorithm consists of different subroutines. The particle encoding/decoding scheme, fitness assignment, and elite preserving scheme are already described in the descriptions on CLPSO and multi-objective optimization. The two other important components of planning are:

(a) *Conductor Size Selection*: The conductor sizes influence both objectives. A heuristic algorithm optimizes these chosen from a given set. An initial load flow is done with random (branch) conductor sizes. Thereafter, conductors rated higher than respective branch flow are assigned randomly to all branches and optimal choices of conductor sizes are found by optimization process.

(b) *Constraint Handling Technique*: Substation and feeder capacities are maintained by incremental capacity addition and proper conductor size selection, respectively. Solutions violating node voltage limits are rejected. The decoding scheme always generates radial networks.

The pseudocodes of the complete algorithm are:

```
Generate initial population;find out non-dominated solutions
and store in elite archive;calculate fitness of particles.
iteration=1
while(iteration<=Max_iteration)
 for i=1,...,Population_size
  {Update velocity & position of the particle as per CLPSO}
  {Decode particle to get the network topology}
  {Select conductor size and perform load flow}
  {Calculate objective functions and particle fitness}
 endfor
 {Find out non-dominated solutions and store in elite archive}
 iteration=iteration+1
endwhile
{Elite archive contains optimal network topologies and
 conductor sizes}
```

6 Simulation Results and Performance Assessment

The proposed algorithm is tested with the data of ([9], [15]). It is a 21-node distribution system with four existing branches (between nodes 1 and 5, node 1 is substation). Nine different conductor sizes are considered. The experiment is carried out to compare the results of SPEA with GA based previous work [9]. The optimum population size (N_{pop}) and maximum iteration ($Iter_{max}$) are obtained. The performances of SPEA and SPEA2 are assessed and compared.

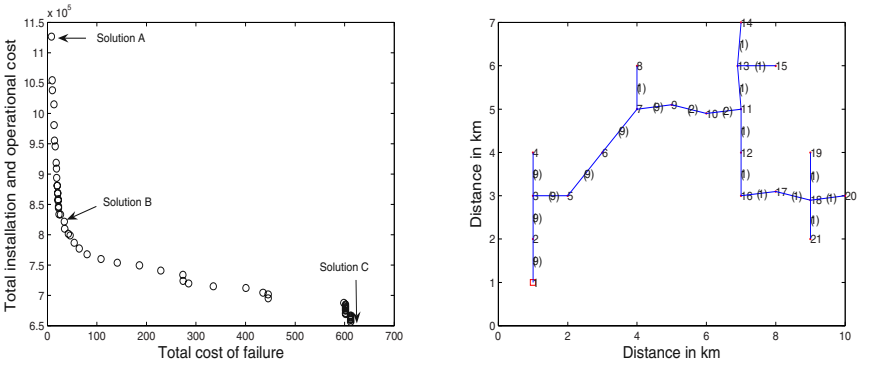
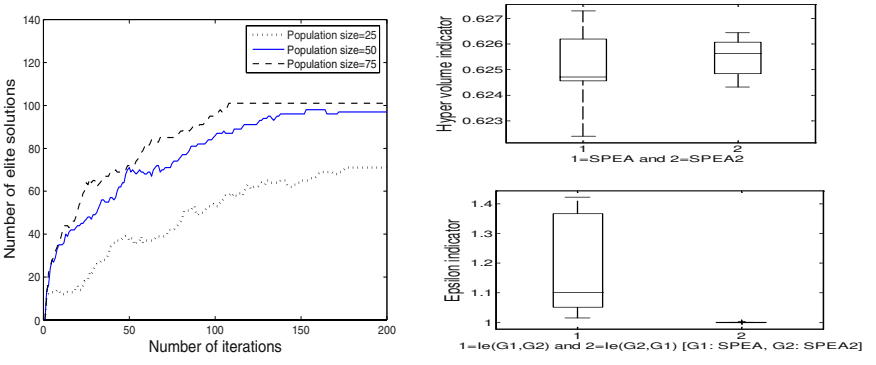


Fig. 2 a. Pareto set approximation, b. One compromised network for 21-node system

One Pareto set approximation ($N_{pop} = 50$, $Iter_{max} = 200$) along with the network of one of the compromised solutions are shown in Fig. 2. The conductor sizes are shown within brackets. A comparison of three solutions, i.e., most economical (A), most reliable (C), and one compromised solution (B) is given in Table 2,

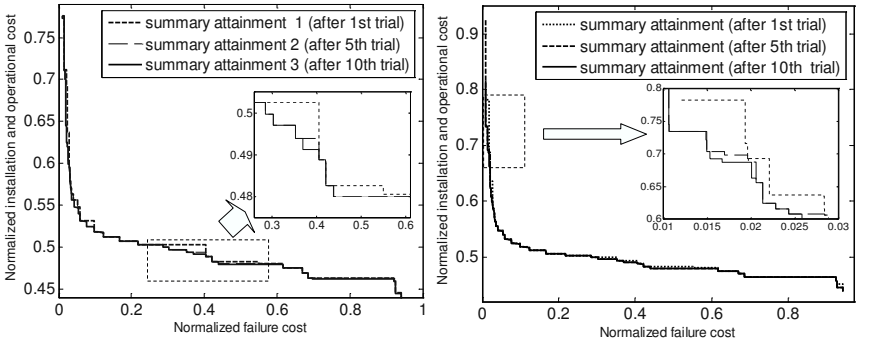
Table 2 Comparison costs between proposed algorithm (with SPEA) and GA [9]

Type	GA [9] (C_{Fi})	CLPSO (C_{Fi})	GA [9] (C_{Fa})	CLPSO (C_{Fa})
Solution A (in dollar)	1.707×10^6	1.1266×10^6	7.7093	8.16
Solution B (in dollar)	8.47×10^5	8.442×10^5	48.06	22.578
Solution C (in dollar)	6.7009×10^5	6.566×10^5	843.1579	612.342



(a) (b)

Fig. 3 a. Elite archive's growth b. (Normalized) boxplots for hypervolume and epsilon indicators



(a) (b)

Fig. 4 Attainment surface plot (a. SPEA and b. SPEA2) after several trials

which shows that CLPSO's performance is comparable with that of GA [9]. To find optimum N_{pop} and $Iter_{max}$, the growth of the elite solutions is observed (Fig. 3a) with no restriction on elite archive size. This shows that the growth saturates within 200 iterations. Cases with $N_{pop} = 50$ and $N_{pop} = 75$ have comparable results. Hence, it is quite reasonable to choose $N_{pop} = 50$ and $Iter_{max} = 200$ as (sub)optimal values.

The performance of any optimizer is assessed in terms of quality of solutions and time required to obtain them. The latter is assessed either by CPU time or maximum number of iterations requirement. In this work, solutions are always obtained within 200 iterations (less than that for GA based algorithm [9]). The solution quality of any stochastic optimizer is measured by statistical hypothesis tests. In multi-objective optimizer, this is carried out by some Pareto compliant quality indicators (unary/binary) and attainment surface plots [16]. The indicators used to assess the performance in this study are as follows:

(a) *Hypervolume indicator* (I_H): It is a unary indicator [16] used to measure the portion of objective space dominated by the approximation set. Thus, a higher I_H signifies better performance. Fig. 3b shows that maximum and minimum value of I_H differ a little (with reference point $(2 \times 10^6, 650)$). This illustrates consistency of the performance. However, SPEA2 has slightly better median value. In this sense, SPEA2 is superior to SPEA.

(b) *Binary epsilon indicator*: It is a binary indicator $[I_\varepsilon(G1, G2)]$ used to measure the comparative performance of two algorithms G1 (SPEA) and G2 (SPEA2) [17]. $I_\varepsilon(G1, G2) = \varepsilon$ such that $z_1 \preceq \varepsilon z_2$ $[z_1 \in G1, z_2 \in G2]$. Algorithm G1 will outperform G2 if $I_\varepsilon(G1, G2) \leq 1 \wedge I_\varepsilon(G2, G1) > 1$. But Fig. 3b illustrates that $I_\varepsilon(G1, G2) > 1$ (median=1.1) and $I_\varepsilon(G2, G1) \leq 1$ (median=1). So, SPEA2 (G2) has better performance.

(c) *Attainment surface plots*: An attainment surface is a boundary comprising of all tightest goals attained by the particular optimizer [16]. After different trials, very little variations in some areas of attainment surface (Fig 4, shown in inset) are obtained. Mostly, these variations are so small that the overall surface plots are indistinguishable. This shows consistency of the outcomes in both types of fitness assignment schemes, i.e., SPEA and SPEA2.

The experimental results demonstrate the efficacy of the proposed algorithm, which is quite comparable with that of GA based algorithm. The statistical hypothesis tests show that the proposed algorithm is capable of generating consistent results in both SPEA and SPEA2. A slightly better performance is achievable in SPEA2.

7 Conclusions

A multi-objective Pareto-based model for electrical distribution system expansion planning has been presented using comprehensive learning PSO to obtain optimum topologies and conductor sizes for the network. The results obtained on a typical distribution system problem are quite noteworthy. The distinct contributions are: problem dimension reduction by a novel encoding/decoding scheme, simultaneous optimization of topology and conductor sizes with minimal non-problem specific heuristics and performance assessment by statistical hypothesis tests. However, more accurate models should incorporate future load uncertainty. These require further investigations.

References

1. Ramirez-Rosado, I.J., Gonen, T.: Pseudodynamic Planning for Expansion of Power Distribution Systems. IEEE Trans. on Power Systems(PWRS) 6(1), 245–254 (1991)
2. Nara, K., Satoh, T., Aoki, K., Kitagawa, M., Yamanaka, M.: Multi-Year Expansion Planning for Distribution Systems. IEEE PWRS 6(3), 952–958 (1991)
3. Vaziri, M., Tomsovic, K., Bose, A.: Numerical Analyses of a Directed Graph Formulation of the Multistage Distribution Expansion Problem. IEEE Trans. on Power Delivery 19(3), 1348–1354 (2004)
4. Nara, K., Kuwabara, H., Kitagawa, M., Ohtaka, K.: Algorithm for Expansion Planning in Distribution Systems Taking Fault into Consideration. IEEE Trans. on Power Systems 9(1), 324–330 (1994)
5. Miranda, V., Ranito, J.V., Proenca, L.M.: Genetic Algorithm in Optimal Multistage Distribution Network Planning. IEEE Trans. on Power Systems 9(4), 1927–1931 (1994)
6. Nahman, J., Spiric, J.: Optimal Planning of Rural Medium Voltage Distribution Networks. Electrical Power and Energy Systems 19(8), 549–556 (1997)
7. Tang, Y.: Power Distribution Systems Planning with Reliability Modeling and Optimization. IEEE Trans. on Power Systems 11(1), 181–189 (1995)
8. Ramirez-Rosado, I.J., Bernal-Agustín, J.L.: Reliability and Costs Optimization for Distribution Networks Expansion Using an Evolutionary Algorithm. IEEE Trans. on Power Systems 16(1), 111–118 (2001)
9. Carrano, E.G., Soares, L.A.E., Takahashi, R.H.C., Saldanha, R.R., Neto, O.M.: Electric Distribution Network Multiobjective Design Using a Problem-Specific Genetic Algorithm. IEEE Trans. on Power Delivery 21(2), 995–1005 (2006)
10. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proc.of IEEE International Conf.on Neural Networks, Pearth, Austrailia, pp. 1942–1948 (1995)
11. Mohemmed, A.W., Sahoo, N.C.: Particle Swarm Optimization Combined with Local Search and Velocity Re-initialization for Shortest Path Computation in Networks. In: Proceeding of IEEE Swarm intelligence Symposium, pp. 266–272 (2007)
12. Liang, J.J., Qin, A.K., Suganthan, P.N., Bhaskar, S.: Comprehensive Learnning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. IEEE Transactions on Evolutionary Computation 10(3), 281–295 (2006)
13. Reyes-Sierra, M., Coello Colleo, C.A.: Multi-Objective Particle Swarm Optimizers: A survey of the state-of-the-Art. International Journal of Computational Intelligence Research 2(3), 287–308 (2006)
14. Deb, K.: Multi-objective Optimization using Evolutionary Algorithms. John wiley and Sons Ltd., Chichester (October 2004); reprinted copy
15. Carrano, E.G., Soares, L.A.E., Takahashi, R.H.C., Saldanha, R.R., Neto, O.M.: Multiobjective genetic algorithm in the design of electric distribution networks: simulation data. Univ. Fed. Minas Gerais, Tech. Rep. (2005), `http://www.mat.ufmg.br/~taka/techrep/agent01.pdf`
16. Knowles, J., Thiele, L., Zitzler, E.: A Tutorial on the Perfomance Assessment of Stochastic Multiobjective Optimizers. Technical Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland (2006)
17. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Fonseca, V.G.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Trans. on Evolutionary Computation 7(2), 117–132 (2003)
18. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Computer Engineering and Networks Laboratory (TIK report-103) (2001)

Prediction of Compressive Strength of Cement Using Gene Expression Programming

Priyanka Thamma and S.V. Barai

Abstract. Gene Expression Programming is employed to predict the 28 days compressive strength of cement mortar. The input parameters considered are C_3S, SO_3, Blaine and Alkali and the output parameter is the 28 days compressive strength. The model was able to predict successfully with a root mean square error of 1.4956. This model is compared with the Fuzzy Logic Model and ANN-GA model. The GEP model is proved to perform better than the Fuzzy Logic Model. It yields an expression that relates the inputs to outputs thereby overcoming the disadvantages of the artificial neural networks.

1 Introduction

Concrete is perhaps one of the most widely used man made materials in the world. Today the construction of various civil engineering marvels around the world is only possible by understanding the behavior of concrete. The behavior of concrete depends on the qualities and quantities of its components. Hence the understanding of the relationships between the qualities and quantities of its components and its properties like strength, workability, durability etc, is very essential. Cement is the most important component of concrete, which determines the quality of concrete to a large extent. The 28 days compressive strength of cement mortar is a function of various chemical parameters like C_3S, C_2S, C_3A, C_4AF and SO_3 contents and physical parameters like Blaine (surface area) and particle size distribution.

Till date many models have been built to predict the compressive strength of cement. A mathematical model has been built in 1994 based on chemical and physical properties of cement following a stepwise linear regression analysis [1]. A Linear Regression model, which included a time sequence dynamic correction procedure, has been earlier built in 1988 relating the strength performance and the cement properties [2]. The year 1994 has seen the development of another statistical model relating the cement setting strength to the chemical composition

Priyanka Thamma · S.V. Barai
Deparment of Civil Engineering, Indian Institute of Technology, Kharagpur 721 302, India
e-mail: priyanka.t29@gmail.com, skbarai@civil.iitkgp.ernet.in

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 203 – 212.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

and surface area [3]. A linearizable mathematical model (AMEBA method) for representing the compressive strength of cement as a time function was proposed in 1998 [4]. Other statistical models like Partial Least Square Analysis Regression apart from ordinary multivariate regression have been employed to predict the 28 days strength [5]. Various statistical approaches were followed in a arriving at the strength equations [6] [7] [8] [9]. The next decade has seen the extensive use of soft computing tools in various engineering applications. Neural networks were used widely in modeling various data. Neural Networks have been employed to determine the strength of unconventional mixes [10][11]. Fuzzy Logic has also been employed in this direction [12]. Genetic Algorithms and Genetic Programming have also been used in the strength prediction [13] [14].

Gene Expression Programming (GEP) that is a development of Genetic Algorithms and Genetic Programming is a very powerful tool in deriving expressions from the data that is input. GEP has also been proved to be better than its predecessor GP (Genetic Programming) [15]. GEP has been used in the past in various civil engineering applications like modeling the deflection basin of flexible highway pavements [16] and also to solve some problems of traffic engineering [17]. GEP has also been used to predict the 28 days cement compressive strength using 19 input parameters, which include the chemical and physical properties along with 1 day; 2 day and 7 days cement compressive strength [18].

This paper presents a limited number of input parameter based GEP model for predicting compressive strength of the cement. For the first time, by just using 4 input parameters, namely C_3S, SO_3, Blaine and Alkali, GEP has been employed in the present paper to predict the 28 days strength. So to arrive at the necessary strength, the expressions obtained are useful in designing the mix. The advantage over the earlier study is that it does not require 1 day, 2 day or the 7 days compressive strength. In the following sections of this paper, model construction for artificial neural networks and fuzzy logic is briefly explained; then the GEP algorithm along with its model construction is described. The obtained results are finally compared and discussed.

2 Data Modeling Tools

Many modeling tools ranging from the simple statistical tools like regression analysis to the more complex soft computing tools like neural networks and genetic algorithms have been used in the past to model the compressive strength of cement as seen in the introduction section. Two soft computing models, namely artificial neural networks and fuzzy logic model, are described in brief in the following sections. These two models are later on compared with the GEP model.

2.1 Artificial Neural Networks

A three layered feed forward type of neural network with 20 inputs and one output was constructed by Sedat Akkurt et al [19]. The 20 inputs included various factors (SiO_2, Al_2O_3 etc…) that determine the compressive strength (output). The data was collected from a cement plant in Izmir, Turkey. A sensitivity analysis done on the

model revealed four important parameters namely % C_3S, % SO_3, % total alkali, and Blaine (cm^2/g) to be the most important in determining the 28 days compressive strength of the cement mortar. Artificial Neural Network was constructed again with 4 input parameters and 1 output parameter as shown in Fig 1.

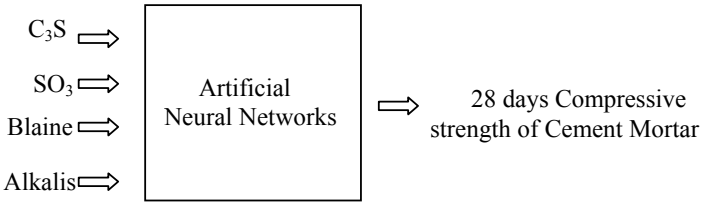


Fig. 1 ANN Model [16]

2.2 Fuzzy Logic Model

A Fuzzy Logic based model was also created by Sedat Akkurt et al [20] to predict the 28 days cement strength. The same inputs parameters as above were used. Triangular membership functions were employed along with prod activator and centroid defuzzification methods. Mamdani type fuzzy rules relating the input variables to the output variables were used in the construction of the model. Though this model is a very useful modeling tool, a large set of fuzzy rule base, a large number of linguistic variables and membership functions make it difficult to understand the relationships between the inputs and outputs. The results obtained using this model are used later for comparison.

3 Background of Gene Expression Programming

GEP is an evolutionary algorithm [21] [22] [23]. It is a development of genetic algorithms and genetic programming. It uses a population of individuals, selects them according to their fitness and luck of roulette wheel, and introduces genetic variation in the individuals using various genetic operators resulting in the development of an expression, which describes the data that is input.

3.1 Initialization of Population

The individuals (population) of GEP comprise of one dimensional arrays formed by the level order traversal of expression trees. An example of such an individual termed as 'gene' is shown in fig 2. A gene consists of two parts namely a head and a tail. A head consists of both functions like +, -, ×, ÷, etc. and terminals. But a tail consists of only terminals. To generate a syntactically correct gene, certain rules are to be followed. The length of the tail should be $h(n-1) + 1$ where h is the length of the head and n is the maximum number of arguments possible for the functions used in the generation of the expression. Two or more genes can be combined to form a chromosome using linking functions like '+', '×', etc…Constants can also be included in the chromosomes but in this paper, constants are not included.[24].

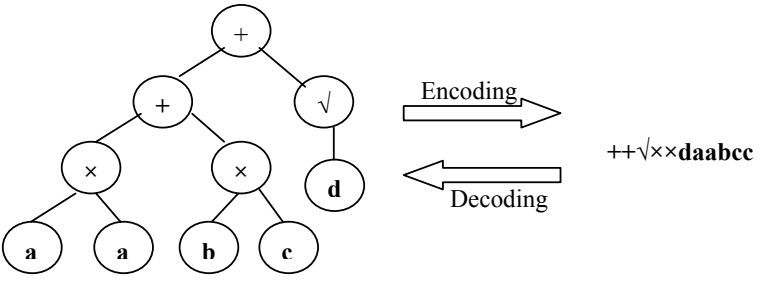


Fig. 2 An example of an expression tree: The gene contains an extra c at the end to make sure that it is syntactically correct. The value of the expression is $y=a^2+b\times c+\sqrt{d}$

3.2 Replication and Selection

After the generation of the population, the individual fitness value is computed using the following expression

$$F = (\sum (M - |C - T|)) \frac{100}{M \times n} \tag{1}$$

Where, M is the range of selection

C denotes the value returned by the target gene
T is the target value
n is the population size

Thus each chromosome has a fitness value. Greater the fitness value, the better it describes the data. The next generation of population is selected using the fitness value of the individuals and the luck of the roulette wheel.

3.3 Genetic Operators

In GEP, there are many genetic operators, like mutation, transposition and recombination, which create genetic variations in the population thereby giving the system a chance to evolve a better population.

Mutation: Mutation is a genetic operator that alters gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With these new gene values, the GEP may be able to arrive at better expression than was previously possible. Mutation helps in preventing the population from stagnating at any local optima. Mutation occurs during evolution according to a user-definable mutation probability.

Transposition: Small fragments of the genes are copied to other places of the chromosomes in this step. Depending on the type of element selected and the position to which they are transposed, three types of transposition are defined. If the transposition involves transposing a fragment to the head of the gene except the root (First position in the head region), then it is termed as transposition of

insertion sequence elements. And if it transposed to the root, then it is called transposition of root insertion sequence element. But in this case the first element of the transposed fragments should be a function whereas in the first case it could either be a function or a terminal. The third type is known as gene transposition in which the entire gene is transposed to the first position and unlike the other forms of transpositions, this transposition deletes the gene that is transposed. The resultant action is only shuffling. This operator is not useful where the linking functions are commutative.

Recombination: Recombination is the simple swapping of a fragment of chromosome between two individual chromosomes. If the swapping occurs at a single point, it is known as one-point recombination. If it occurs at two points, it is known as two point recombination. If an entire gene is swapped, then it is termed as gene recombination. All the three types of recombination are used in the algorithm.

All the genetic operators result in syntactically correct genes. After introducing the genetic variations in the population, the fitness of the chromosomes are once again computed. If the fitness is not satisfactory, then selection and replication are carried on. Genetic operators are employed and then fitness is computed again. These iterations, as shown in Fig 3, are carried on until satisfactory results are obtained.

3.4 GEP Algorithm

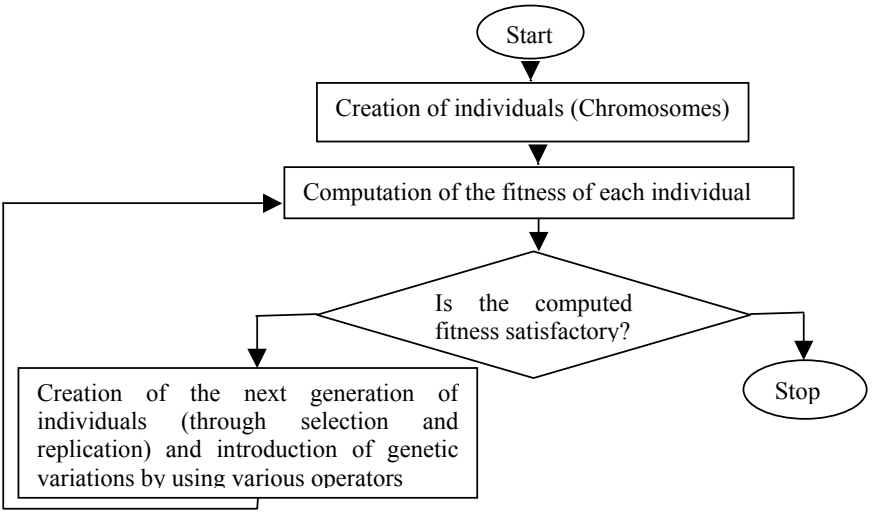


Fig. 3 Algorithm used in GEP model

4 Data Collection

The 50 data points (Tables 1 and 2) are taken from a paper [20] for training/testing.

Table 1 Data used for training the GEP model

S.no	Strength (N/mm^2)	C_3S (%)	So_3 (%)	Blaine (cm^2/g)	Alkali (%)
1	53.9	54.0	3.0	3530	1.1
2	51.9	54.8	2.9	3680	0.9
3	50.8	64.6	2.6	3850	1.0
4	54.5	56.9	2.7	3580	0.8
5	50.4	61.3	2.3	3780	0.9
6	58.4	62.4	2.8	3590	0.9
7	54.8	64.6	2.5	4090	0.8
8	51.8	59.3	2.8	3500	1.1
9	51.3	61.8	2.7	3630	1.1
10	54.7	61.3	3.0	3580	1.0
11	54.1	60.4	2.6	3680	1.0
12	54.5	55.6	3.1	3510	1.0
13	52.1	63.1	2.6	3540	0.9
14	54.2	55.6	2.7	3620	0.9
15	53.8	67.3	2.6	4020	0.8
16	51.5	58.7	3.0	3550	0.9
17	48.9	65.4	2.3	3730	0.9
18	53.2	58.0	2.7	3420	1.0
19	54.7	65.0	2.5	4070	0.8
20	54.3	62.0	2.9	3720	1.0
21	52.5	61.4	2.7	3840	0.9
22	51.3	63.5	2.5	3540	1.0
23	51.1	62.8	2.3	3580	0.9
24	54.1	62.8	3.0	3750	1.1
25	53.5	58.9	3.0	3540	1.0
26	53.6	62.3	2.5	3910	0.9
27	55.4	57.7	2.7	3480	1.0
28	53.7	55.8	3.1	3420	0.9
29	55.2	60.7	2.8	3740	1.1
30	55.5	59.3	2.5	3750	1.1
31	55.6	60.7	3.0	3840	0.9
32	52.1	63.2	2.5	4010	0.9
33	51.6	59.3	2.6	3450	1.0
34	53.0	65.8	2.6	4050	0.9
35	50.5	57.4	2.5	3390	1.1
36	54.0	62.0	2.4	3490	1.0
37	52.1	59.7	2.2	3890	1.0
38	53.6	61.7	2.4	3630	0.9
39	53.0	63.6	2.8	3680	0.9
40	53.5	61.6	2.8	3630	1.1
41	49.9	64.9	2.4	3900	1.0
42	54.2	61.0	2.8	3700	0.9

Table 2 Data used for testing the model

S.no	Strength (N/mm^2)	C_3S (%)	So_3 (%)	Blaine (cm^2/g)	Alkali (%)
1	53.9	57.3	2.8	3560	1.0
2	55.4	62.3	2.8	3640	0.9
3	51.5	62.4	2.5	3590	1.1
4	49.8	60.8	2.2	3520	1.1
5	55.6	55.9	2.8	3620	1.0
6	53.8	56.8	2.7	3620	1.0
7	52.5	56.4	3.0	3370	1.1
8	51.7	61.2	2.7	3610	0.9

5 Model Construction and Results

Using the GEP algorithm shown in Fig 3 and the data in Table 1, expressions were developed. The ranges of the parameters employed are tabulated in Table 3. The expressions thus obtained are given in Table 4.

Table 3 Range of the parameter values used in GEP modeling

Head size	6,7
Population size	50
Number of genes	3-7
Number of iterations	100-1000
Mutation probability	0.04-0.06
IS transposition rate	0.2-0.3
IS transposition length	3,4
RIS transposition rate	0.2-0.3
RIS transposition length	3,4
1 point Recombination rate	0.2-0.35
2 point recombination rate	0.2-0.35
Gene Recombination rate	0.2-0.35

In the GEP expressions, the following notation has been adopted:

y = compressive strength; a = C_3S; b = SO_3; c = Blaine; d = Alkali

The number of iterations carried out for each model is different.

Where, RMSE stands for Root Mean Square Error and MEA for Mean Absolute Error.

Fig 4 and 5 show a comparison of the three models – GEP model, Fuzzy Logic model [20] and ANN-GA model [19].

Expression 6 in Table 5 has been used in evaluating the GEP model. Expression 6 is an ensemble of expression 2, 3 and 4. For the training set, GEP model has an RMSE of 1.7948 and Fuzzy Logic model has an RMSE of 1.8453. The same trend continues for the testing set too. The former predicts with an

Table 4 Developed Expressions

S. No	Expressions	Training		Testing	
		RMSE	MEA	RMSE	MEA
1	$y = \frac{\sin a}{b} + e^{d} + \sqrt{c} - b - \sqrt{a} + d$	2.4988	1.9603	1.8622	1.5535
2	$y = 3b + \frac{a}{\log(\frac{a}{b})} + \sin(\log(b))$	2.1557	1.8046	1.8935	1.5925
3	$y = 2d + \sqrt{\sqrt{c^2 \log(b)}} + \sin(d) + \log(b)$	2.0576	1.6293	1.8345	1.4784
4	$y = 10\sqrt{\sqrt{\ln(c)}} + \cos(c(\cos(a-c))) + b + \frac{b}{\ln(c)} + 1$	2.0335	1.6235	1.8755	1.4557
5	y=(exp 3 + exp 4)/2	1.829	1.4256	1.5729	1.2060
6	y = (exp 2 + exp 3 + exp 4)/3	1.7948	1.3986	1.5280	1.2778
7	$y = 2b + d + \sin(d) + \log(c) + e^{\log(c)} + e^{d}$	1.8016	1.4002	1.6174	1.4386
8	y = (exp 3 *(exp 7)^{3})$^{1/4}$	1.7701	1.3620	1.4956	1.3146
9	Fuzzy model [17]	1.8453	1.4119	1.8010	1.6125
10	ANN-GA model [16]	1.7756	1.3310	1.2835	1.0250

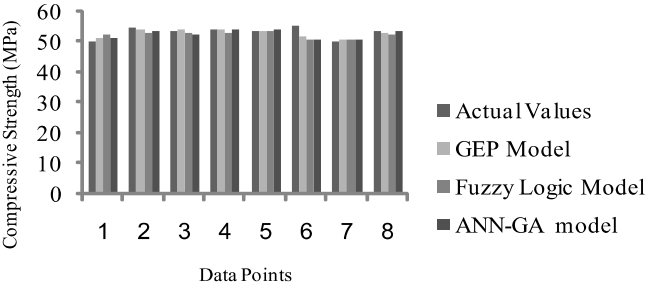


Fig. 4 Comparison of the three models using training data

RMSE of 1.5280 and the latter with an RMSE of 1.8010. This shows that the GEP model predicts better than the Fuzzy logic model. But the ANN-GA model predicts the compressive strength of cement with a slightly less error compared to the GEP model. But ANN-GA model is a black box. Hence GEP is proven to predict the cement compressive strength better than the other two.

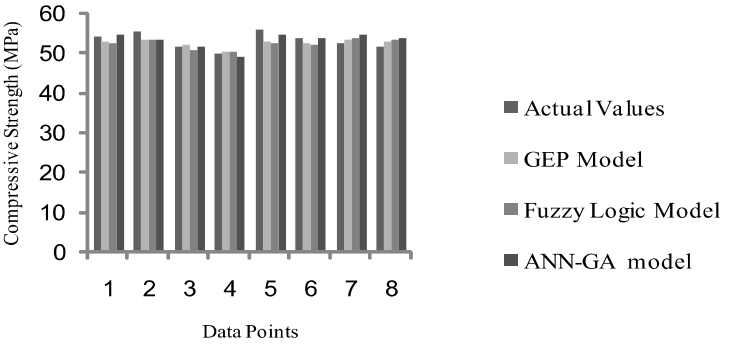


Fig. 5 Comparison of the three models using testing data

6 Summary and Future Work

Present study shows that GEP is a very convenient tool in modeling the compressive strength of cement mortar. Unlike ANN, GEP is not a black box. Given the parameter values, the compressive strength can be predicted by just using a calculator. The GEP model has been compared to the Fuzzy Logic Model and has proved to be better than the Fuzzy Logic Model. The ensemble equations were observed to yield better results compared to the original expressions developed by the GEP model. A larger data set is preferable for a better understanding. Hence, a system in which genetic algorithms can be employed to find out the weights of the individual expressions (to be used in ensembling) can be modeled.

References

1. Tsivilis, S., Parissakis, G.: A Mathematical model for the Prediction of Cement Strength. Cem. Concr. Res. 25(1), 9–14 (1995)
2. Relis, M., Ledbetter, W.B., Harris, P.: Prediction of Motor Cube Strength from Cement Characteristics. Cem. Concr. Res. 18, 674–686 (1988)
3. Zhang, Y.M., Napier-Munn, T.J.: Effects of Particle Size Distribution, Surface area and Chemical Composition on Portland Cement Strength. Powder Technology 83, 245–252 (1995)
4. Tango, C.E.: An Extrapolation method for Compressive Strength Prediction of Hydraulic Cement Products. Cem. Concr. Res. 28(7), 969–983 (1998)
5. Swinning, K., Hoskuldusson, A., Justnes, H.: Prediction of compressive strength up to 28 days from microstructure of Portland cement. Cem. Concr. Res. 30(2), 138–151 (2008)
6. Douglas, E., Pouskouleli, G.: Prediction of Compressive Strength of Mortars made with Portland cement- Blast furnace slag- fly ash Blends. Cem. Concr. Res. 21, 523–534 (1991)
7. Das, S.K., Yudhdir: A simplified model for prediction of pozzolanic characteristics of fly ash based on chemical composition. Cem. Concr. Res. 36, 1827–1832 (2006)

8. Zelic, J., Rusic, D., Krstulovic, R.: A mathematical model for prediction of compressive strength in cement-silica fume blends. Cem. Concr. Res. 34, 2319–2328 (2004)
9. Hwang, K., Noguchi, T., Tomosawa, F.: Prediction model of compressive strength development of fly-ash concrete. Cem. Concr. Res. 34, 2269–2276 (2004)
10. Sebastia, M., Olmo, I.F., Irabien, A.: Neural network prediction of unconfined compressive strength of coal fly ash-cement mixtures. Cem. Concr. Res. 33, 1137–1146 (2003)
11. Stegemann, J.A., Buenfeld, N.R.: Prediction of unconfined compressive strength of cement paste containing industrial wastes. Waste Management 23, 321–332 (2003)
12. Topcu, I.B., Saridemir, M.: Prediction of Compressive Strength of concrete containing fly ash using artificial neural networks and fuzzy logic. Computational Materials Science 41, 305–311 (2008)
13. Lim, C.H., Yoon, Y.S., Kim, J.H.: Genetic algorithm in mix proportioning of high-performance concrete. Cem. Concr. Res. 34(3), 409–420 (2004)
14. Chen, L.: Study of Applying Macro evolutionary Genetic Programming to Concrete Strength Estimation. J. Computing in Civil Engineering 17(4), 290–294 (2003)
15. Wu, C.S., Huang, L., Kang, L.S.: The automatic Modeling of complex functions based on Gene Expression Programming. In: Fourth International Conference on Machine Learning and Cybernetics, vol. 5, pp. 2870–2873 (2005)
16. Terzi, S.: Modeling the Deflection Basin of Flexible Highway Pavements by Gene Expression Programming. Journal of Applied Sciences 5(2), 309–314 (2005)
17. Bagula, A.B.: Traffic Engineering Next Generation IP Network using Gene Expression Programming. Network Operations and Management Symposium 10, 230–239 (2006)
18. Baykasog˘lu, A., Dereli, T., Tani, K.: Prediction of cement strength using soft computing techniques. Cem. Concr. Res. 34(11), 2083–2090 (2004)
19. Akkurt, S., Ozdemir, S., Tayfur, G., Akyol, B.: The use of GA-ANNs in the modeling of compressive strength of cement mortar. Cem. Concr. Res. 33(7), 973–979 (2003)
20. Akkurt, S., Tayfur, G., Can, S.: Fuzzy logic model for the prediction of cement compressive strength. Cem. Concr. Res. 34(8), 1429–1433 (2004)
21. Ferreira, C.: Gene Expression Programming in Problem Solving. In: 6th Online World Conference on Soft Computing in Industrial Application (2001), http://www.gene-expression-programming.com/webpapers/GEPtutorial.pdf
22. Ferreira, C.: Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. Complex Systems 13(2), 87–129 (2001)
23. Gene Expression Programming, http://www.gene-expression-programming.com
24. Ferreira, C.: Function Finding and the Creation of Numerical constants in Gene Expression Programming. In: Advances in Soft Computing: Engineering Design and Manufacturing, pp. 257–266. Springer, Heidelberg (2003)

Fault-Tolerant Nearest Neighbor Classifier Based on Reconfiguration of Analog Hardware in Low Power Intelligent Sensor Systems

Kuncup Iswandy and Andreas König

Abstract. An increasing interest of low-power integrated intelligent sensor systems is intended for the efficient realization of mobile and distributed realizations. Wireless sensor networks (WSN) are one possible example, where long term sensor vigilance and data acquisition and rather sporadic and brief communication phases occur. Mixed-signal realization, in particular exploiting sub-threshold implementation are interesting, but suffer from susceptibility to environmental and process parameter deviations. Redundancy and reconfiguration based on evolutionary approaches can overcome these problems and raise the yield. In this paper, the behavioral model of a previously implemented one-nearest neighbor (1-NN) reconfigurable mixed-signal classifier is modified by using Gaussian distribution of process parameter deviations, which is more close to real problems rather than uniform distribution used in our prior work. An eye-tracking example will be employed for the case study. To compensate classifier performance due to static deviations, the prototypes are adjusted or reconfigured by Particle Swarm Optimization (PSO) and successfully proved in our simulation results. The yield could be increased considerably, so that PSO and instance based reconfiguration gives broader applicability to low-power mixed-signal circuits.

1 Introduction

Integrated sensor systems for intelligent problem solutions find increasing industrial interests and applications. Intelligent ventilation or airbag-triggering systems from the automotive domain are just two established examples. In particular, power consumption is of paramount importance due to restricted energy resources for mobile and distributed realizations. Even state-of-the-art energy harvesting techniques still

Kuncup Iswandy · Andreas König
Institute of Integrated Sensor Systems, University of Kaiserslautern,
67663 Kaiserslautern, Germany
e-mail: {kuncup, koenig}@eit.uni-kl.de

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 213–222.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

demand for extremely power economic circuit and system implementations. Wireless sensor network, robotics, or Ambient Intelligence applications, such as Assisted Living tasks, are common examples of application fields. Mixed-signal solutions, in conjunction with dedicated device and circuit technology exploiting sub-threshold operation of MOS transistors, has been the subject of low-power designers for more than two decades. Compact and efficient mixed-signal low-power solutions can be achieved for intelligent integrated sensor systems. However, these are extremely vulnerable to environmental and process parameter deviations. Recognition or classification rates of intelligent systems could be substantially compromised, effectively reducing the circuit and system yield to unacceptable levels.

One of the established techniques for yield optimization, which is adopted in our work, is to reconfigure the circuit, employing internal redundancy and reconfiguration options, to achieve an instance specific correction or in-spec behavior. The first time such an approach was industrially exploited, was with the Intel ETANN analog neural network chip, where chip-in-the-loop-learning (CHILL) was applied, adjusting EEPROM synapses by gradient descent learning [8]. In more recent approaches from the field of evolvable hardware, evolutionary computing (genetic algorithms, GA) were applied to restore circuit performance or breed new circuits [9].

In the work presented here, we pick up results and complete a task from a previous project, where the development of a design methodology for very low-power mixed-signal implementations of intelligent sensor systems has been pursued. One of the benchmark examples was eye-shape detection and tracking for human-machine-interfacing in 3D-display control [5]. Vision sensors with high dynamic range and focal-plane low-power feature computation as well as a mixed-signal classifier were designed and implemented[1]. On behavioral level, employing the proprietary QuickCog system, application systems, e.g., eye shape detectors, where developed eliminating redundancies on algorithmic level with corresponding power savings. System components, e.g., the classifier, were designed in Cadence DFW II with different circuit solutions and validation took place by sending data and classifier contents to Cadence DFW II simulations using HDLs as a vehicle for export and result import. Thus, consistent behavior of circuits and systems could be assured and/or optimization could come in.

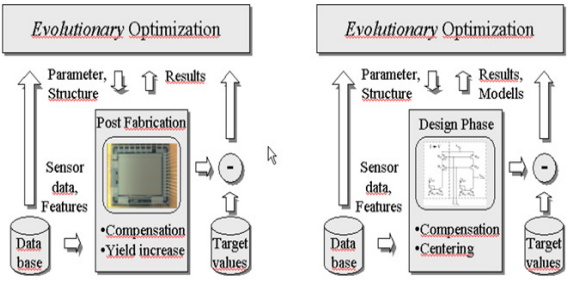


Fig. 1 Reconfiguration approach for intelligent sensor circuit and system (left) chip (right) circuit-level reconfiguration

[1] Designers were J. Skribanowitz, J. Dge, C. Mayr, T. Bormann, and C. Klug.

In addition to these design time activities, restoration options of manufactured chips were investigated. Figure 1 explains the concept, which is close to CHILL and today's evolvable hardware approaches. In the next section, the physical classifier is explained, that is subject of the case study by PSO. Then PSO will be briefly rehearsed and the applied behavioral model be explained. The experimental section explains details of the employed data, assumed perturbations and the achieved results. Concluding, an outlook on potential improvements and future exploitation will be given.

2 Reconfigurable Hardware Implementation of Nearest Classifier

Nearest neighbor classification is a well established technique. The basic *k*-NN approach suffers for embedded application from the need to store all training samples. Iterative reduction algorithms, e.g. the reduced-nearest neighbor technique of Gates [5], alleviates this problem, requiring only the storage of a fraction of the prototypes. On the other hand, established techniques of feature selection can reduce the data dimension, too. Finally, implementation of friendly city block metric can be applied in the required 1-NN classification. Figure 2 shows the layout and manufactured chip of a low-power analog classifier, that contains two modules for eight prototypes and eight features, each. The analog circuits implement Manhattan distance and a looser takes all circuit (LTA) for class affiliation due to minimum distance prototype. Prototypes are digitally stored in 6-bit RAM cells and converted by local DACs. The manufactured circuit potentially is very vulnerable to process deviations, which potentially can be compensated by shifting prototype vectors, i.e., changing or reconfiguring the 6-bit patterns in the chip to restore classification accuracy and, thus, yield. In this work, PSO will be investigated for its aptness to achieve the aspired compensation. For reasons of flexibility with regard to investigated perturbations, a behavioral model of the chip has been established, that will also be discussed in the following section.

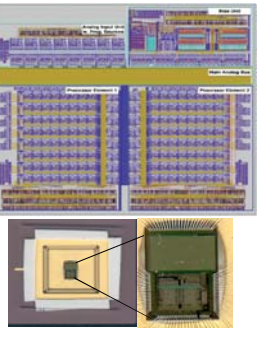


Fig. 2 Layout and chip of reconfigurable mixed-signal classifier chip

3 Optimization of Reconfigurable Prototypes

3.1 Particle Swarm Optimization

Particle swarm optimization (PSO) method for function optimization is one of the evolutionary computation techniques and was originally introduced by Kennedy and Eberhart [2]. PSO is a population based technique to explore through the multi-dimensional search space. For our application, the particles represent a set of the prototypes put into a row vector as shown below

$$\mathbf{x}_i = (x_{i11}, x_{i12}, ..., x_{i1D}, x_{i21}, ..., x_{iPD}), \tag{1}$$

where P is the number of prototypes and D indicates the number of features.

The particles evaluate their positions relative to a fitness function. The fitness function is determined by an application-specific objective function (i.e., 1-NN classifier chip with the modeling of deviation parameters, see Section 3.2).

In every iteration, the velocity vector of each particle is updated based on sharing memories of their best positions, where there are three parts, the momentum part, the cognitive part, and the social part. The original formula developed by Kennedy and Eberhart was improved by Shi and Eberhart [6] by adding new parameter (inertia weight) into the original PSO algorithm, where we adopt their formula in this paper. The particles are manipulated according to the following equation

$$\mathbf{v}_i^{t+1} = \omega \cdot \mathbf{v}_i^t + c_1 \cdot \mathbf{r}_1 \cdot (\mathbf{L}_i^t - \mathbf{x}_i^t) + c_2 \cdot \mathbf{r}_2 \cdot (\mathbf{G}_i^t - \mathbf{x}_i^t), \tag{2}$$

and

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}, \tag{3}$$

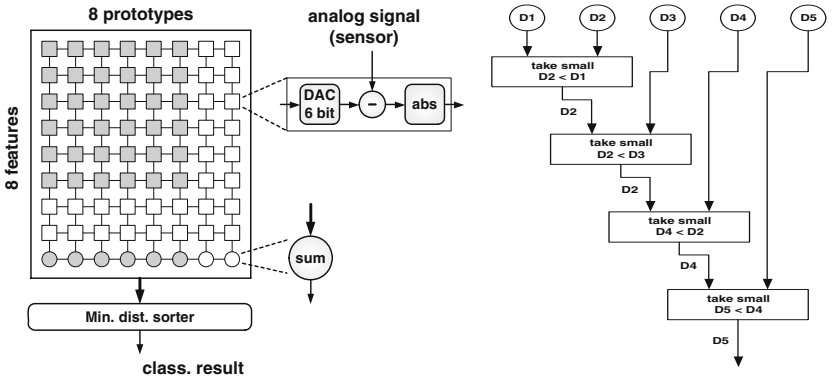
where L is the local best position of each particle, G is the global best particle found among all the particles. The parameter c1 and c2, called the cognitive and social learning rates, are constant values set in range [0.5, 2.5]. The parameter ω, used to control the influence of the previous velocity [7], is in range of [0.4, 1].

3.2 Objective Function and Gaussian Model

The classifier chip described in Section 2 is modeled in C-code. Prototypes are quantized to 6-bit and sensor data comes in as 'analog' unquantized values. Figure 3 shows the computational steps or operators of 1-NN classifier with Manhattan distance modeled in C. The perturbation pattern for operators, representing manufacturing problems, was introduced using Gaussian distribution for representing the individual deviation characteristics of the hardware. Perturbation for operators was modeled by a scale and an offset parameter as follows

$$Z_{new} = Z_{old} \cdot (1 + R) + h, \tag{4}$$

$$R = randn() \cdot \sigma_R + \mu_R, \tag{5}$$



(a) Structure (b) distance sorter

Fig. 3 Block diagram of 1-NN classifier model

$$h = randn() \cdot \sigma_h + \mu_h, \tag{6}$$

where R is an additive value of scaling factor, which affects the magnitude of output and h is an offset value. Both scaling and offset variables are generated randomly using Gaussian distribution (5) and (6), where the standard deviation parameter of Gaussian distribution is summarized in Table 1. Both parameters μ_R and μ_h is set to zero. Three operators, i.e. DAC, subtraction, and absolute, are illustrated in Fig. 3(a). For a decision making with regard to the minimum distance, a sequential comparator is applied in the model as shown in Fig. 3(b), where the inputs of each comparator are modeled in the same way with other operators.

Table 1 Deviation values used for generating Gaussian distribution

Scaling	R_0	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
σ_R	0.00	0.02	0.04	0.06	0.08	0.10	0.12	0.14	0.16	0.18	0.20
offset	h_0	h_1	h_2	h_3	h_4	h_5					
σ_h	0.00	0.01	0.02	0.03	0.04	0.05					

3.3 Optimization Procedure

The procedure of adjusting prototypes is as shown in Fig. 4. The raw sensor data set is split into training and test sets. The feature computation is applied to extract important features of the sensor signal. The three processes, i.e. Automatic Feature Selection (AFS), min-max normalization, and Reduced-Nearest-Neighbor (RNN), are sequentially employed by QuickCog [5, 4]. After the required prototypes are obtained, they will be adapted by PSO for compensation using the hardware modeling

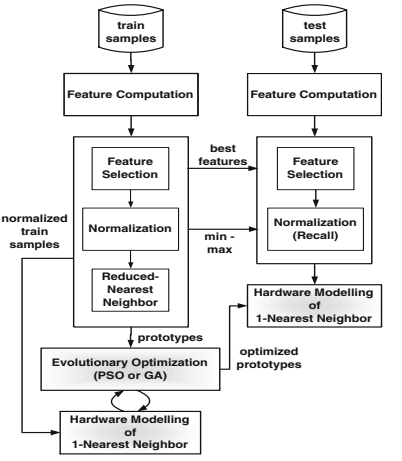


Fig. 4 Procedure of prototype optimization

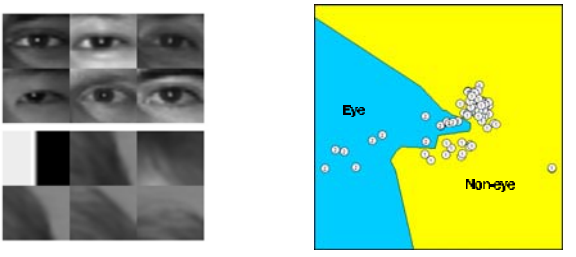
of 1-NN. Then, the adjusting prototypes will be reused for generalization using test set.

The parameter settings of PSO in our experiments were determined as following:

- Population size is 20.
- The inertia weight ω is initially set to 1 and linearly decreased to 0.4 in the end of iteration.
- Both c_1 and c_2 are set to 2.
- Maximum iteration is limited to 100.
- The speed of particles in moving to new place is limited in range of [-1, 1].
- The vector space of particles are set in range of [0, 1].

4 Experiments and Results

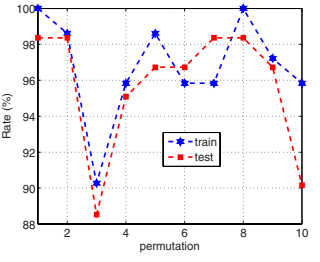
In our experiment, eye-shape detection benchmark data with eye (28 samples) and non-eye (105 samples), is employed. This data set was split into training and testing sets. The group of eye was equally split for training and generalization, whereas the group of non-eye was 58 samples for training and 47 for generalization. The examples of eye and non-eye images are shown in Fig. 5(a). The projection of data analysis using nonlinear mapping technique [3] can be shown in Fig. 5(b). This eye data has 12 features extracted by using Gabor filter (feature computation) [5]. The AFS with sequential backward selection (SBS) reduced to six features. For selecting more robust and unbiased features, AFS can be improved by using cross-validation techniques [1]. A RNN using Manhattan distance computed six prototypes [5], which consist of three samples from eye class and other three samples from non-eye class. In the ideal case (without perturbation) of classifier, the classification accuracy is 100%.

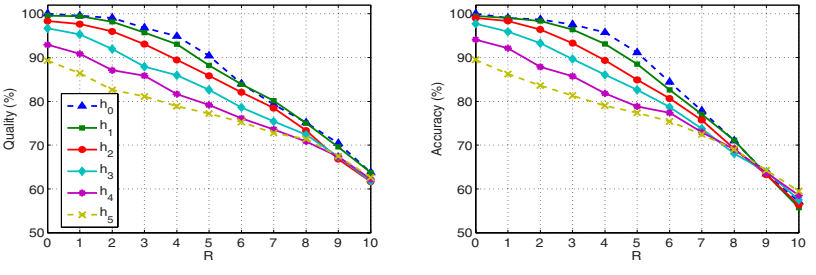


(a) (b)

Fig. 5 Six eye and six non-eye examples of images (left) and projection of eye-shape feature space (right)

Fig. 6 Performance of varying storage positions of prototypes



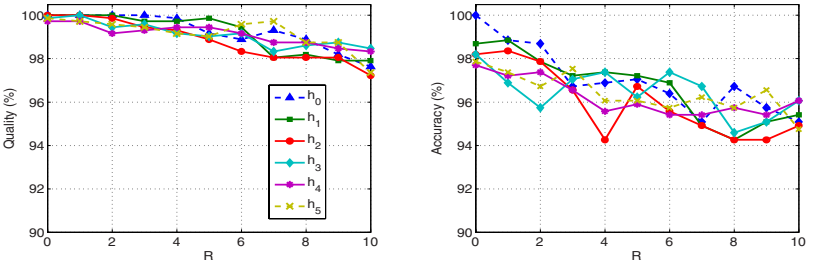


(a) Train (b) Test

Fig. 7 Average of quality and accuracy rate without PSO

In the first experiment, the sensitivity of storage position of prototypes was investigated and the results were shown in Fig. 6. The model was generated by Gaussian perturbation with $\sigma_R = 0.1$ and $\sigma_h = 0.01$. The permutation was done 10 times. From this experiment, we can increase the performance of classifier by finding the best storage position of prototypes.

Next, we investigated the sensitivity of the classifier with regard to the variation of Gaussian parameter (given in Table 1) for scaling R and offset h variables, which express the circuit deviation. In this experiment, we kept with the same storage



(a) Train (b) Test

Fig. 8 Average of quality and accuracy rate adjusted by PSO

position of original prototypes. We generated 10 samples for each alteration of σ_R and σ_h. The results of training and test sets are summarized in Fig. 7, where the performance of chip classifier degrade when both σ_R and σ_h parameters are increased. In the recovering process, we computed the average of 10 runs of each perturbation case. Figure 8 shows that, the adjusting original prototypes by PSO can increase the performance of chip classifier above 97% of quality value and achieve above 94% of recognition rate. We scaled up curves in Fig. 8 for observation purpose.

As an indicator for category of acceptable chip classifier, we defined a threshold of minimum value, i.e., 95% of classification rates. For Gaussian perturbation with $\sigma_R = 0.1$ and $\sigma_h = 0.03$, we analyzed 10 modeling classifiers. We found only three samples performing above the threshold and seven samples failed. With reconfiguring the prototypes by PSO, the performance of seven samples were restored to exceed the threshold value and another good samples were boosted to attain better performance. For perturbation with $\sigma_R = 0.2$ and $\sigma_h = 0.05$, we analyzed in similar way by taking also 10 samples and found all 10 samples failed. The PSO increased the performance of classification rates for all samples, but only eight samples can be categorized as good samples. Figure 9 summarizes these simulation and shows the efforts of restoration by PSO. In addition, we analyzed the sensitivity (True Positive Ratio) and specificity (True Negative Ratio) of these 10 samples, where PSO can improve the modeling classifiers with high sensitivity and high specificity. The analysis results were shown in Table 2. Simulating chip manufacturing problems with two different Gaussian perturbation parameters, we proved the advantage of readjusting prototypes of classifier by PSO on 100 samples of the modeling classifier. In Table 3, we categorized 100 samples into five groups, i.e. G1 = 100%, G2 = (100%,95%], G3 = (95%,90%], G4 = (90%,70%], and G5 = (70%,0%], with regard to their performance the simulation results shows that adjusting prototypes by PSO with Gaussian perturbation ($\sigma_R = 0.1$ and $\sigma_h = 0.03$) can give 95 of 100 samples performed above the threshold value, compared to 48 samples for original prototypes. For high perturbation, PSO can give 78 of 100 samples compared to only 6 samples without optimized by PSO.

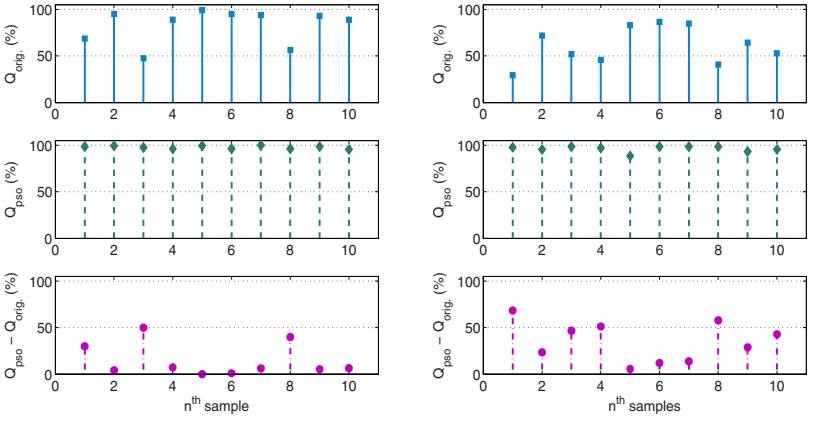
Table 2 Sensitivity and specificity of 10 model 1-NN classifier chips

Model	Original		PSO	
	Sensitivity	Specificity	Sensitivity	Specificity
$\sigma_R = 0.1$, $\sigma_h = 0.03$	0.927[a] (0.130[b])	0.802 (0.243)	0.906 (0.074)	0.996 (0.005)
$\sigma_R = 0.2$, $\sigma_h = 0.05$	0.800 (0.285)	0.559 (0.312)	0.862 (0.160)	0.987 (0.012)

[a] Mean value.
[b] Standard deviation.

Table 3 Simulating of 100 model 1-NN classifier chips and comparison between adjusted (A) and original (B) prototypes

Model	G1		G2		G3		G4		G5	
	A	B	A	B	A	B	A	B	A	B
$\sigma_R = 0.1$, $\sigma_h = 0.03$	14	0	81	48	5	26	0	15	0	11
$\sigma_R = 0.2$, $\sigma_h = 0.05$	3	0	75	6	19	25	3	35	0	34



(a) $\sigma_R = 0.1$ and $\sigma_h = 0.03$ (b) $\sigma_R = 0.2$ and $\sigma_h = 0.05$

Fig. 9 Restoring performances of 10 samples by PSO

5 Conclusion

The paper deals with the importance and potential problems of mixed-signal low-power integrated sensor system implementations. Reconfiguration was proposed to cope with manufacturing problems and achieve acceptable yield. PSO was investigated for that purpose based on a behavioral model of an implemented 1-NN classifier and found highly effective, even for extreme deviations. The experimental results showed that the reconfigurable approach can cope with different Gaussian perturbation parameters. However, the approach still lack to restore all chip samples

above the threshold value due to exhausted resources of prototypes. To restore all samples to achieve above the threshold value, we can increase the degree of freedom in our optimization technique by adding more prototypes and finding the sequence of prototype storage positions. In future work, more sophisticated optimization and other classifier models, e.g., SVM implementations, will be regarded.

Acknowledgment

The funding of DFG (Ko-, SPP 1062 VIVA) , of part of this work is gratefully acknowledged.

References

1. Iswandy, K., König, A.: Towards effective unbiased automated feature selection. In: 6th Int. Conf. on Hybrid Intelligent Systems (HIS 2006) (2006)
2. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
3. König, A.: Interactive Visualization and Analysis of Hierarchical Neural Projections for Data Mining. IEEE Trans. Neural Networks 11, 615–624 (2000)
4. König, A., Gratz, A.: Advanced Methods for the Analysis of Semiconductor Manufacturing Process Data. In: Pal, N.R., Jain, L.C. (eds.) Advanced Techniques in Knowledge Discovery and Data Mining, pp. 27–74. Springer, Heidelberg (2005)
5. König, A., Mayr, C., Bormann, T., Klug, C.: Dedicated implementation of embedded vision systems employing low-power massively parallel feature computation. In: Proc. of the 3rd VIVA-Workshop on Low-Power Information Processing, pp. 1–8 (2002)
6. Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: Proc. of the IEEE Int. Conf. on Evolutionary Computation, pp. 69–73 (1998)
7. Shi, Y., Eberhart, R.C.: Parameter selection in particle swarm optimization. In: Proc. of the 7th Annual Conf. on Evolutionary Programming, pp. 561–600 (1998)
8. Tam, S.M., Gupta, B., Castro, H.A., Holler, M.: Learning on an analog vlsi neural network chip. In: IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 701–703 (1990)
9. Tawdross, P., König, A.: Mixtrinsic multi-objective reconfiguration of evolvable sensor electronics. In: Proc. of 2nd NASA/ESA Conf. on Adaptive Hardware and Systems, pp. 51–57 (2007)

Text Documents Classification by Associating Terms with Text Categories

V. Srividhya and R. Anitha

Abstract. Automatic text categorization has always been an important application and research topic since the inception of digital documents. With the prevalence of digital documents and the wide use of e-mail and web documents, text categorization is regaining interest and is becoming a central problem in digital text collections. There have been many approaches to solve this problem, mainly from the machine learning community. This paper explores the use of association rule mining in building a text categorization. This approach has the advantage of a very fast training phase, and the rules of the classifier generated are easy to understand and manually tuneable. The investigation leads to conclude that association rule mining is a good and promising strategy for efficient automatic text categorization.

1 Introduction

Amazing development of Internet and digital library has triggered a lot of research areas. Text categorization is one of them. Text categorization is a process that group text documents into one or more predefined categories based on their contents [1]. It has wide applications, such as email filtering, category classification for search engines and digital libraries.

Basically there are two stages involved in text categorization. Training stage and testing stage. In training stage, documents are preprocessed and are trained by a learning algorithm to generate classifier. In testing stage, a validation of classifier is performed. There are many traditional learning algorithms to train data, such as Decision trees, Naïve-Bayes (NB), Support Vector Machines, k-Nearest Neighbor (kNN), Neural Network (NN), etc [9]. Nowadays, text categorization becomes fundamental given the large number of on-line documents that have to be sorted and grouped. For example large companies could use text classifiers for incoming e-mail triage and memo categorization. Text classifiers can be used to classify web pages, in-coming emails, memos, news and any other text collection. Building a text classifier usually necessitates a training set consisting of a

V. Srividhya
Lecturer, Avinashilingam University for Women, Coimbatore-43
e-mail: vidhyavasu@gmail.com

R. Anitha
Director, Department of M.C.A, K.S.R. College of Technology, Tiruchengode-637 209

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 223 – 231.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

collection of text documents already associated with topical categories. Once a classifier is built with the training set, a test set, consisting of documents with known categories, is classified and the found class labels compared to the existing categories to determine the effectiveness of classifier.

This paper exploits the use of association rules mining in building categorization system from relatively large training set. The remainder of the paper is organized as follows: Section 2 gives an overview of related work in text categorization and association rule mining. Section 3 shows a new categorization approach. Experimental results are described in section 4. Summarization of research and discussion of future work are narrated in Section 5.

2 Related Work

Many text classifiers have been proposed in the literature using machine learning techniques, probabilistic models, etc. Although a lot of approaches have been proposed, automated text categorization is still major area of research. The use of association rule mining for building classification models is very new. This classification systems discovers the strongest association rules in the database and use them to build categorizer.

In the following subsections a more detailed overview of the related work is presented from both domains: text categorization and association rule mining.

2.1 Text Categorization

In the past decade, great attention was paid to the text categorization problem Most of the text classifiers that were developed and proposed are either machine learning based or statistical based. Classifiers based on probabilistic models have been proposed starting with the first presented in literature by Maron in 1961 and continuing with naïve Bayes [6] that proved to perform well. ID3 and C4.5 are well-known packages whose cores are making use of decision trees to build automatic classifiers [4, 5]. K-nearest neighbor (k-NN) is another technique used in text categorization [12]. Another method to construct a text categorization system is by an inductive learning method. This type of classifier is represented by as set of rules in disjunctive normal form that best cover the training set [7]. As reported in [11] the use of bigrams improved text categorization accuracy as opposed to unigrams use. In addition, in the last decade neural networks and support vector machine (SVM) were used in text categorization and they proved to be powerful tools [10].

2.2 Association Rule Mining

2.2.1 Association Rules Generation

Association rule mining has been extensively investigated in the data mining literature. Many efficient algorithms have been proposed. Among the proposed algorithms the best known are apriori and FP-Tree growth [2]. Association rule mining typically aims at discovering associations between database items in a transactional. Given a set of transactions D = {T1,.., Tn} and a set of items I= {i1,.., im} such that any transaction T in D is a set of items in I, an association rule is an

implication A-->B where the antecedent A and the consequent B are subsets of a transaction T in D, and A and B have no common items. For the association rule to be acceptable, the conditional probability of B given A has to be higher than a threshold called minimum confidence. Association rules mining is normally a two-step process, wherein the first step frequent item sets are discovered (i.e. item-sets whose support is no less than a minimum support) and in the second step association rules are derived from the frequent item-sets.

2.2.2 Associative Classifiers

Besides the classification methods, associative text categorization, and a new method that builds associative general classifiers. In this case the association rule mining represents the learning method. Two models were presented in the literature: CMAR and CBA [8].

The main idea behind this approach is to discover strong patterns that are associated with class labels. The next step is to take advantage of these patterns such that a classifier is built and new objects are categorized in the proper classes.

3 Building an Associative Text Classifier

A new document categorization algorithm was proposed by M.Antonie and Osmar R.Zaiane [3]. It has the following advantages: it makes no assumption of term independence and it is fast during both training and categorization. ARC-BC is an Apriori based algorithm that only interested in rules that indicate a category label. In this algorithm each set of documents that belong to one category is considered as a separate text collection to generate association rules. If a document belongs to more than one category, this document will be present in each set associated with the categories that the document falls into.

3.1 Association Rule Generation

In this algorithm apriori algorithm to discover frequent term-sets in documents. Eventually, these frequent item sets associated with text categories represent the discriminate features among the documents in the collection. The association rules discovered in this stage of the process is further processed to build the associative classifier. Here consider each set of documents belonging to one category is considered as a separate text collection to generate association rules from. If a document belongs to more than one category this document will be present in each set associated with the categories that the document falls into.

There are two approaches have been considered in building an associative classifier. The first one ARC-AC (Association Rule-based Classifiers with All categories) is used to extract association rules from the entire training set. As a result of discrepancies among categories in a text collection of a real world application, discovered that it is difficult to handle some categories that have different characteristics (small categories, overlapping categories or some categories having documents that are more correlated than others). As a result ARC-BC (Association

Rule-based Classifier By Category) is used to solve such problems. The ARC-BC algorithm is described more detail in Figure 1 below.

Algorithm: ARC-BC Find association rules on the training set of the text when the text corpus is divided in subsets by category. **Input:** A set of documents of documents (D) of the form $D_i = \{C, t_1, t_2, .t_n\}$ where C_i is the category attached to the document and, t_n are the selected terms for the document; min_support is the threshold for support. **Output:** Association rules like: $t_1 \wedge t_2 \wedge \ldots\ldots t_n \Rightarrow C_i$; t_j is a term and C_i is the category.

```
Pseudo Code:
(1) C1 = {Candidate 1 term-sets and their support}
(2) F1 = { Frequent 1 term-sets and their support}
(3) for ( i = 2; Fi-1 != ∅; i++) {
(4) Ci = candidates generated from Fi-1
(5) Di = FilterTable (Di-1 , Fi-1)
(6) foreach document d in Di {
(7) foreach c in Ci {
(8) c.support += Count(c, d)
(9) }
(10) }
(11) Fi = { c ∈ Ci | c.support > min_support}
(12) }
(13) Sets = Ui { c ∈ Fi | i > 1}
(14) foreach itemset I in Sets{
(15) R += { I => Category}
(16) }
```

Fig. 1 ARC-BC Algorithm

In step 2, it generates the frequent 1-itemset. In steps 3-12, it generates all the k-frequent item sets. In step 14-16, it generates the association rule. It's almost same as the Apriori, but there are some differences: 1) In step 5, the filter table function removes the terms not in frequent i-1 sets, which are not useful in the next loop. 2) In step 14-16, it generates rules by combining the frequent item sets with category.

Table 1 presents a set of rules that are discovered in the text collection. Such rules are composing the classifier. The rules are human readable understandable. Although the rules are similar to those produced using a rule-based induced system, the approach is different.

Table 1 Example of association rules composing the classifier

People^editor^dear => Letter
People^disease^medical=>Health
Players^matches =>Sports

3.2 Prediction of Classes Associated with New Documents

The set of rules that were selected represent the actual classifier. This categorizer will be used to predict to which classes new documents are attached. Given a new document, the classification process searches in this set of rules for finding those classes that are the closest to be attached with the document presented for categorization. This subsection discusses the approach for laelling new documents based on the set of association rules that forms the classifier.

Given a document to classify, the terms in the document would yield a list of applicable rules. If the applicable rules are grouped by category in their consequent part and the groups are ordered by the sum of rules confidences, the ordered groups would indicate the most significant categories that should be attached to the document to be classified. This order category is named as dominance factor δ. The dominance factor allows us to select among the candidate categories only the most significant. When δ. is set to a certain percentage a threshold is computed as the sum of rules confidences for the most dominate category times the value of the dominance factor. Then, only those categories that exceed this threshold are selected. The function (Take K Classes) selects the most k significant classes in the classification algorithm. The algorithm for the classification of a new object is given in Figure 2.

Algorithm: Classification of a new object

Input: A new object to be classified o; the Associative classifier (ARC); the dominance factor δ; the confidence threshold T

Output: Categories attached to the new object

```
(1) s <- ∅ /*set of rules that match */
(2) foreach rule r in ARC (the sorted set of rules)
(3) if (r ⊂ o) {count++}
 (4) if (count == 1)
(5) fi.conf <- r.conf /*keep the first rule confidence*/
(6) S <- S Ù r
(7) else if (r.conf > fi.conf-T)
(8) S <- S Ù r
(9) else exit
(10) divide S in subsets by category: S1,S2…Sn
 (11) foreach subset S1,S2…Sn
(12) sum the confidences of rules and divide by the number of rules in Sk
(13) if it is single class classification
(14) put the new document in the class that has the highest confidence sum
(15) else /*multi-class classification*/
(16) Take K Classes (S, δ)
(17) assign these k classes to the new document
```

Fig. 2 Classification of a new object

4 Experimental Result

4.1 Experiment Data

Data source of experiments is news document. The total number of documents is 7347, each document with a pre assigned category label. Stop words are used as filter and then words are stemmed by porter stemmer. After preprocessing, there are unique terms stored in the database. The corpus is split into two parts, 5000 document training set and 2347 document testing set respectively.

4.2 Experimental Results and Analysis

Table 2 shows the result of generated rules with different training time based on three support thresholds 75%, 80%, 85%.

Table 2 Generated rules and Training time

Data set size	1000 documents		2000 documents		3000 documents		4000 documents		5000 documents	
Support %	N	TT (Sec)	N	TT (Sec)	N	TT (Sec)	N	TT (Sec)	N	TT (Sec)
75	07	1.69	21	1.79	30	2.33	29	2.89	26	3.60
80	03	1.62	20	1.59	22	2.21	22	2.69	18	3.42
85	03	1.52	12	1.13	17	2.11	16	2.31	14	2.48

N - # of Generated Rules; TT – Training Time (sec).

From Table 2, firstly we can see that the number of generated rules varies with the size of training set. From 1000 to 3000 documents, the number of rules increases correspondingly, and for 5000 documents, it decreases comparing with the 3000 documents. Secondly, when the support thresholds are going up (75%, 80%, and 85%), the number of rules and training time are reduced. Figure 3 shows the chart of dataset vs. training time compared by two support values, whose data is derived from the above table , the training time with higher support value 85%(bottom line) has a better performance, and noticed that when the size of data set is greater than 3000, the difference of training time is remarkably increased. The trend was clearly showed in Figure 4.

The validation of the association rule classifier is implemented and table 3 shows the experiment result. Two testing data sets 100 and 1000 are applied on two training data sets with three support thresholds: 75%, 80% and 85%. The validation is measured by accuracy which is the percentage of correctly classified data. The accuracy has a significant increase when the support threshold is set up to 85% for the 5000 data set. The difference shows that increasing the number of

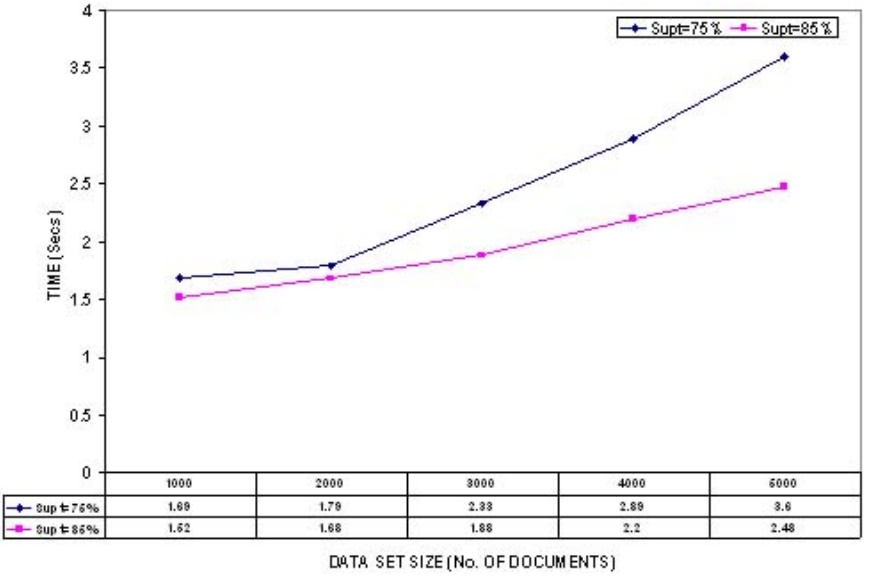


Fig. 3 Dataset vs Training Time (Supt=75% and 85%)

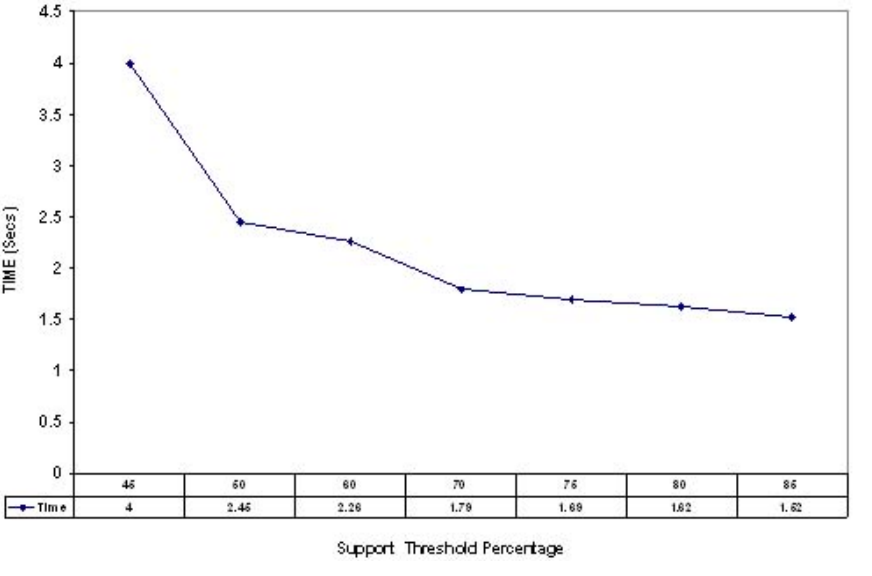


Fig. 4 Training Time vs. Support (Data set Size = 1000)

files in training set and the support threshold can increase the accuracy classification.

Table 3 Accuracy of classifier for different support and data set

Number of Documents in Training Set	Number of documents in Testing Set	Support		
		75%	80%	85%
1000	100	43%	42%	52%
	1000	49%	54%	53%
5000	100	57%	55%	78%
	1000	54%	51%	79%

5 Conclusion and Future Work

This paper introduced a new technique for text categorization. It employs the use of association rules. Our study provides evidence that association rule mining can be used for the construction of fast and effective classifiers for automatic text categorization. One major advantage of the association rule based classifier is that it does not assume that terms are independent and its training is relatively fast. Furthermore, the rules are human understandable and easy to be maintained. Feature selection can be done adding the weight of each term in the documents and pruning the terms with lower weight. The feature selection will reduce the number of terms as well as reduce the noisy of the terms.

References

[1] Aas, K., Eikvil, A.: Text categorization: A survey, Technical report, Norwegian Computing Center (June 1999)
[2] Agrawal, R.S.: Fast Algorithm for Mining Association Rules. In: Proc. VLDB Conf. Santiago, Chile (1994)
[3] Antonie, M.L., Zaiane, O.R.: Text document categorization by term Association. In: Proc. of the IEEE International Conference on Data Mining (ICDM 2002), pp. 19–26 (2002)
[4] Cohen, W., Hirsch, H.: Joins that generalize: text classification using whirl. In: 4th International Conference on Knowledge Discovery and Data Mining (SigKDD 1998), New York City, USA (1998)
[5] Cohen, W., Singer, Y.: Context-sensitive learning methods for text categorization. ACM Transactions on Information Systems (1999)
[6] Lewis, D.: Naive (bayes) at forty: The independence assumption in information retrieval. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398. Springer, Heidelberg (1998)
[7] Li, H., Yamanishi, K.: Text classification using esc-based stochastic decision list. In: 8th ACM International Conference on Information and knowledge Management, Kansas City, USA (1999)
[8] Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: ACM Int. Conf. on Knowledge Discovery and Data Mining, New York City, NY (1998)

[9] Yingbo, M., Gang, W., Zheyuan, Y., Xin, S.: The implementation of text categorization with term association, Project Report (2003)
[10] Ruiz, M., Srinivasa, P.: Neural networks for text cargorization. In: 22nd ACM SIGIR International Conference on Information Retrieval, Berkeley, CA, USA (August 1999)
[11] Tan, C.M., Wang, Y.F., Lee, C.D.: The use of bigrams to enhance text categorization. Journal of Information Processing and Management (2002)
[12] Yang, Y.: An evaluation of statistical approaches to text categorization, Technical Report CMU-CS-97-127, Carnegie mellon University (April 1997)

Applying Methods of Soft Computing to Space Link Quality Prediction

Bastian Preindl, Lars Mehnen, Frank Rattay, and Jens Dalsgaard Nielsen

Abstract. The development of nano- and picosatellites for educational and scientific purposes becomes more and more popular. As these satellites are very small, high-integrated devices and are therefore not equipped with high-gain antennas, data transmission between ground and satellite is vulnerable to several ascendancies in both directions. Another handicap is the lower earth orbit wherein the satellites are usually located as it keeps the communication time frame very short. To counter these disadvantages, ground station networks have been established. One input size for optimal scheduling of timeframes for the communication between a ground station and a satellite is the predicted quality of the satellite links. This paper introduces a satellite link quality prediction approach based on machine learning.

1 Background

Within the last decade the educational and academic approaches in space science made huge steps forward. Driven by the development of small satellites for taking scientific or educational payload of any kind into space, universities all over the

Bastian Preindl
Institute of Analysis and Scientific Computing, Vienna Technical University, Austria
e-mail: bastian@preindl.net

Lars Mehnen
Institute of Analysis and Scientific Computing, Vienna Technical University, Austria
e-mail: mehnen@technikum-wien.at

Frank Rattay
Institute of Analysis and Scientific Computing, Vienna Technical University, Austria
e-mail: frank.rattay@tuwien.ac.at

Jens Dalsgaard Nielsen
Department of Electronic Systems, Aalborg University, Denmark
e-mail: jdn@es.aau.dk

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 233–242.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

world started to design, develop and launch small satellite projects based on the Cubesat standard [1] [2].

Small satellites often operate in the low earth orbit (LEO), which leads to a very high orbit frequency. As a consequence the communication timeframe between satellite and affiliated ground station tends to be about 30 minutes a day whereas the ground station is idle for the remaining time [3]. As the available time for communication between satellite and mission control can be crucial for a mission, investigations have to take place to optimize the usage of ground stations and to significantly extend the time a satellite can communicate with the mission control center.

A sophisticated approach for a world-wide interconnection of independent ground stations [4] is the Global Educational Network for Satellite Operations (GENSO) [5]. Its aim is to share ground station capacity to different mission controls by creating a hybrid, supervised peer-to-peer network using the Internet.

Interconnecting a very large amount of satellite ground stations will form the base for novel scientific approaches in the domain of link quality determination and prediction, overall optimization of space up- and downlinks as well as hardware utilization and the influence of environmental conditions on space communication. This will be a major step forward for research as well as for educational purposes.

2 Specific Optimization Aims

The Global Educational Network for Satellite Operations constitutes the scientific base for a multitude of research fields and investigations. The research aims on the identification and utilization of link quality information.

For the first time in history the possibility is provided to gain focussed information about the quality of LEO satellite downlinks from a huge amount of independent ground stations throughout the world. The gained information can be processed and applied in various ways, whereas the optimization of the network itself is only one of them.

The research aims rely on a broad base on the GENSO project as it delivers the needed raw data for the majority of investigations and offers the possibilities for applying the discovered novel models and algorithms in a productive and mature system. It also provides an unprecedented availability of ground stations which makes such high sophisticated investigations even possible. The majority of the research results and derived models will directly flow back into the project as implementations forming sophisticated cornerstones of GENSO. In case of soft-computing approaches a continuous learning, refinement and adaption process takes place as the resulting model is dynamic and self-optimizing.

Each of these specific aims constitutes a novel approach in its field of science and can support space operations upcoming the next decade and probably far beyond. All of the topics during research are aimed on focus on the reduction of resource and energy consumption to enhance the outcome of space missions drastically.

2.1 Space Up- and Downlink Quality Identification and Determination

As a base for all further scientific approaches in this direction an overall metric has to be identified for measuring and comparing the quality of satellite links. This is due to the application of different protocols, modulations and frequency bands on one hand and the operation in a heterogeneous hardware environment on the other hand. Figure 1 illustrates the complexity of communication between ground and space.

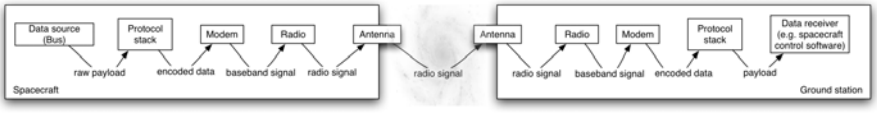


Fig. 1 The communication signal path between a spacecraft and a ground station

As the network itself is not able and also not permitted to be aware of the content of the transferred data between a mission control center and a spacecraft a method has to be identified to measure the quality in a completely passive way.

The investigations include a possible design for the determination of uplink quality for future application.

2.2 Identification of Correlations between Environmental Conditions and Space Link Quality

The collected information about satellite link quality can be set into relation with environmental data to identify correlations between environmental conditions and their impact on link quality. The ground station network can therefore be utilized as very large distributed sensor cluster.

Environmental variables which will be taken into account cover:

- *Space weather*: Solar bursts, ion storms and similar space weather conditions do have a heavy impact on space communication links. Large ground-bounded sensor clusters and satellite missions for measuring space weather provide detailed information about the current situation in space.
- *Earth weather*: Rain, humidity, snow and other well-known earth weather conditions also do affect radio communication. Not only the weather on ground but also the conditions within the stratosphere, mesosphere and ionosphere have to be taken into account.
- *Atmospheric effects* : Atmospheric gases in different layers can have impact on radio links whereas higher frequencies are more susceptible. Effects like ice crystals in higher atmospheric layers are taken into account.

- *Geographical circumstances*: The positions of sun and moon play an important role amongst other geographical and temporal circumstances in the condition of radio links. Therefore their relative positions have to be taken into account.

Actually not being an environmental variable but having perhaps the largest influence on link quality are communication parameters like carrier frequency, bandwidth, applied modulation and encodings, filters and many others.

The resulting calculation model establishes the base for further investigations on environmental impacts on one hand and quality predictions on the other hand.

2.3 The Long-Term Impact of Climate Changes on Short-Range Satellite Communication

Based on the derived model of dependencies between environmental conditions and the quality of satellite links the focus is set on the impact of specific magnitudes as they play a possible role within the global climate change. The different impacts on the different radio frequency bands, modulations and encodings are investigated and a long-term impact prediction on satellite communication based on current prognoses is developed.

The outcoming result is supposed to be a significant factor in future spacecraft and ground segment design as a possible outcome of the investigations identifies susceptible communication variables in respect of the global climate change. This also has to be taken into consideration when designing commercial satellites which stay operational in orbit for a decade or more.

2.4 The Rapid Determination of Spacecraft Orbital Elements

When new spacecrafts are exposed into their designated orbit by a launch vehicle, their exact position and even the orbit is not known. It can take up to several days until institutions like the North American Aerospace Defense Command (NORAD) have clearly identified the new spacecraft in orbit and provide the exact orbital elements (the Keplerian Elements) defining the position and attenuation of an object in space. The worst situation is that no communication can be established to the spacecraft during that time. The first hours and days in orbit are the most important ones since most problems occur during that time [6]. A significant amount of space mission failures have been a consequence of no communication possibilities between ground station and spacecraft causing enormous losses of investment, both time and resources.

Taking these circumstances and the fact that institutions like NORAD are under governmental control into consideration points to the need for an independent, reliable process for rapid orbit determination.

An algorithm shall be proposed, modeled, simulated and approved for the automated orbit determination by utilizing the ground stations participating in GENSO

and the model derived from former quality considerations. The algorithm is intended to reduce the time for gaining a precise orbit from several days to a couple of hours and therefore significantly support future space mission.

2.5 The Short-Term Prediction of Space Communication Link Quality

Based on the aggregated information of satellite link quality and current environmental conditions a short-term prediction model will be designed. Not only the useful booking of ground stations is going to be optimized but also the probability of successful space links is dramatically raised.

This raises the amount of retrievable satellite data during nominal operation significantly and can even play a major role in the success of a whole mission in critical situations wherein time is the most important factor.

2.6 Automated Identification of Imprecise Ground Stations

By reversing the model for short-term prediction of the link quality and investigating on differences between the predicted and effective link quality after the pass of a spacecraft has taken place, ground stations with receiving and transmitting capabilities below the predicted level can be identified.

The ground station operators are informed about problems with their communication hardware which offers the possibility of having a low-cost calibration facility for non-commercial ground stations. In parallel the ground stations are downgraded within the network to avoid the use of broken or imprecise ground stations in critical situations.

2.7 Automated Determination of Spacecraft Health and Orbit Changes

Based on the model for prediction of the satellite link quality anomalies in a spacecraft's behavior and its health status can be (indirectly) identified and countermeasured. Not only signal weaknesses and anomalies can be identified but also orbit deviations. The orbit information can be automatically re-adjusted by applying a prediction algorithm as for the rapid orbit determination after a spacecraft launch.

3 Data Mining Architecture

The machine learning pipeline has to consist of various subsystems in order to form a comprehensive environment for the selection and application of modern AI

classifiers to achieve all of the projected aims. The subsystems obtain and preprocess all possible variables which could influence communication links on the one hand and provide the prediction results as input to various decision making problems and for error recognition on the other hand. Figure 2 visualizes the order and interdependencies of the specific subsystems.

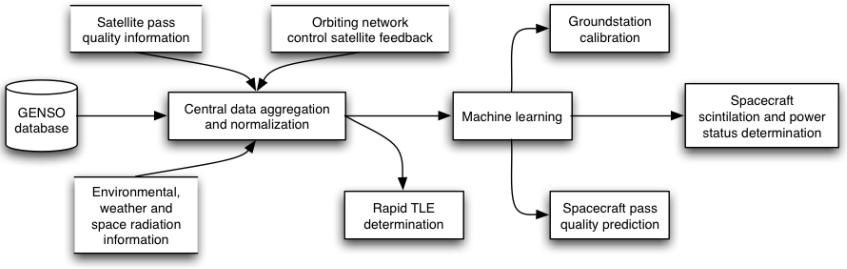


Fig. 2 The components of the machine learning pipeline

The purposes of the specific subsystems are:

- *GENSO database*: Information about the involved hardware (spacecraft and ground station) is a prerequisite for precise data normalization. The network provides this information to participating ground stations and mission controls for scheduling and communication purposes. The feature vector for each classification is fed with hardware and communication details using the provided interfaces to the GENSO database.
- *Satellite pass quality information*: The quality of a satellite pass is measured at the communication ground station, normalized and delivered to the core server, where it is added to the feature vector for learning. These measurements constitute one of the cornerstones of the prediction model derivation. As the satellite link quality is the feature to be classified, it is only provided for compiling training sets, test sets and for model calibration.
- *Environmental influences, weather, and space radiation information*: Current data about the environmental conditions during a measured pass has to be collected utilizing web mining technologies. The collected environmental data forms another cornerstone of the prediction model derivation.
- *Orbiting network control satellite feedback*: A novel approach for non-commercial satellites having a very high impact on nowadays and future space missions is the first orbital space link measurement instrument, a satellite whose main purpose is to send and receive predefined test data to actively measure a space link's quality. The resulting information about bit error rates (BER), the most significant quality metric of a digital communication link in both sending and receiving direction, is highly accurate and outperforms any passive link determination by far. As soon as the link quality information from the satellite(s) is available to the GENSO network it will complement and partially substitute the passively collected quality information from the ground stations.

- *Central data aggregation and normalization*: The data collected from several sources needs to be aggregated and normalized. Heterogeneous hardware environments have to be taken into account just as well as satellite orbits and their relative position to measuring ground stations, amongst several other particularities like daytime, solar and lunar eclipses etc. The outcome of this process is a high amount of data properly preprocessed for the application of different model derivation methods. It therefore forms the base for training and testing different classifiers.
- *Rapid orbit (TLE) determination*: Based on the normalized data retrieved primarily from local ground station measurements an algorithm can be identified and approved by simulation to optimize the determination of spacecraft orbits.
- *Model derivation based on machine learning methods*: Having the aggregated and normalized data as foundation, classification of former learning features are applied to determine a model for attribute interdependencies. The derived model is continuously calibrated during the operation of the network.
- *Ground station calibration*, *spacecraft pass quality prediction* and *spacecraft scintillation and power status determination*: Conclusions on these subjects can be drawn from the measured pass quality after a pass has taken place by selecting other features than the pass quality as classification attribute.

The primary value to be predicted and therefore the classification value is the link quality itself, as it is returned from the ground station network as satellite pass quality information. The definition of link quality is much more complex as expected and further investigations have been undertaken by the authors to identify a comparable and at the same time expressing size for satellite link quality in [7]. The satellite pass quality information as it is derived from the network delivers raw signal strength meter readings from the ground station radio hardware as a function in time, whereas noise and signal strength is not separated.

The reason for not choosing a more explicit value for representing the link quality in the first step is that e.g. the bit error rate (BER) requires knowledge about the transmitted bit sequence in advance for a bit-for-bit comparison if no sophisticated forward error correction (FEC) is implemented in the communication protocol as for example in DVB-S [8]. In case of the most common protocol in non-commercial satellite communication, AX.25 [9], only a simple cyclic redundancy checksum algorithm is applied (CCITT-16) and therefore no detailed bit error rate information can be obtained. Amongst this the payload transmitted and received by ground stations is also random to a certain extend (except for protocol headers) what makes a bit-for-bit comparison impossible.

To gain a numerical value as input size for the classification algorithm, data aggregation and normalization has to be applied on the sequence of data representing the signal strength at a specific time. The requested numerical value is the BER, which has the big advantage that it is comparable with the data provided by the orbiting network control satellite [10].

Together with the orbital elements which precisely describe an object's position in space also the location of the ground station is derived from the network's core

database amongst several other network and ground station specific parameters. These parameters can be used to set the measured link quality in relation to the satellite's elevation and therefor its distance to the ground station. [11] has proven that the bit error rate is directly interconnected to the satellite elevation. As a consequence the derived BER has to be normalized with respect to the distance of the spacecraft.

The myriad of values collectable about ground weather, space weather, environmental conditions and many other, continuously changing, influences on space links has to be collected, normalized and set in a relation with the transformed satellite link quality and the network information to build the foundation for the application of machine learning classifiers.

4 Prediction Model Development

Inspecting the data types of the input parameters, their amount and dimensions and the data type of the classification results allows a preliminary prediction of which kind of learning algorithm will deliver the best quality on the test sets. As classified training data is provided, supervised learning is applied. Supervised learning generates a global model that maps the input feature vector to classifications. After parameter adjustment and learning, the performance of the algorithm will be measured on a real data test set that is separated from the training set and by later on integrating the classification model into GENSO.

The feature vector consists of mainly numerical values, for example equivalent isotropically radiated power (EIRP), current spacecraft tumble rate, antenna gain, frequency band, baud rate, angle between antenna pointing direction and sun, moon and horizon, distance to spacecraft, solar wind activity, humidity and temperature on ground, atmospheric ion gas concentration, longitude and latitude, air pressure, and several more. The prime classification attribute, the link quality, is also numerical, but will be at first divided into quality classes. When more precise BER information, most likely from the measurement satellite, is available, the quality will be predicted as a continuous value using regression. This requires a machine learning algorithm capable of dealing with numerical classes respectively regression. The authors expect Support Vector Machines respectively Support Vector Regression to deliver a very good classification performance [12]. For training and testing the machine learning algorithm collection and workbench RapidMiner (formerly YALE) [13] will be utilized. The different classifier performance will be evaluated using 10-fold cross-validation.

The size of the provided training data and validation test set depends on the amount of ground stations participating in GENSO and being able to evaluate the link quality. Every satellite pass constitutes one attribute vector. A LEO satellite passes a ground stations horizon 6-8 times a day [3] which results in 6-8 pass reports per ground station per satellite per day. In its public Beta phase GENSO is expected to interconnect approximately 30 ground stations and track 15 satellites

which results in more than 3000 data sets collected each day. Hence the network will provide an enormous amount of satellite pass reports and a more than adequate amount for training and validation in the first 24 hours of operation.

5 Conclusion

Applying methods of artificial intelligence and non-linear optimization techniques to scheduling input parameters of a highly dynamic distributed cluster of satellite ground stations can lead to a significant increase in mission return from all spacecrafts in non-geostationary orbit.

Student space projects, non-profit communities like the radio amateurs and probably even commercial space missions will strongly benefit from this work in various aspects: higher reliability, improved resource utilization and the establishment of quality assurance for satellite space links.

References

1. Toorian, A., Diaz, K., Lee, S.: The cubesat approach to space access. In: IEEE Aerospace Conference, 2008, NASA Jet Propulsion Lab., Pasadena, CA (2008)
2. Klofas, B., Anderson, J., Leveque, K.: A Survey of CubeSat Communication Systems. Technical report, California Polytechnic State University and SRI International (2008)
3. Cakaj, S., Keim, W., Malarić, K.: Communications Duration with Low Earth Orbiting Satellites. In: Proceedings of the 4th IASTED International Conference on Antennas, Radar and Wave Propagation (2007)
4. Tuli, T.S., Orr, N.G., Zee, R.E.: Low Cost Ground Station Design for Nanosatellite Missions. In: AMSAT Symposium 2006, San Francisco (2006)
5. Preindl, B., Page, H., Nikolaidis, V.: GENSO: The Global Educational Network for Satellite Operations. In: Proceedings of the 59th International Astronautical Conference, Glasgow, UK, International Astronautical Federation (2008)
6. Chouraqui, S., Bekhti, M., Underwood, C.I.: Satellite orbit determination and power subsystem design. In: Proceedings of 2003 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2003, vol. 7, pp. 4590–4592 (2003)
7. Preindl, B., Mehnen, L., Nielsen, J.D.: Measuring satellite link quality in ground station networks with heterogenous hardware environments. Technical report, Vienna Technical University and Aalborg University, Denmark (2008)
8. Arsinte, R.: Effective Methods to Analyze Satellite Link Quality Using the Build-in Features of the DVB-S Card. Acta Technica Napocensis -Electronics and Telecommunications 47(1), 33–36 (2006)
9. Parry, R.R.: AX.25 [Data Link Layer Protocol For Packet Radio Networks]. IEEE Potentials 16(3), 14–16 (1997)
10. Preindl, B., Mehnen, L., Nielsen, J.D.: Design of a Small Satellite for Controlling a Ground Station Network. Technical report, Vienna Technical University, Austria and Aalborg University, Denmark (2008)
11. Yang, C.-Y., Tseng, K.-H.: Error rate prediction of the low Earth orbit (LEO) satellite channel. In: IEEE International Conference on Communications, ICC 2000, vol. 1, pp. 465–469 (2000)

12. Hussain, S., Khamisani, V.: Using Support Vector Machines for Numerical Prediction. In: IEEE International Multitopic Conference, INMIC 2007, December 2007, pp. 1–5 (2007)
13. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: Rapid Prototyping for Complex Data Mining Tasks. In: Ungar, L., Craven, M., Gunopulos, D., Eliassi-Rad, T. (eds.) KDD 2006: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 935–940. ACM, New York (2006)

A Novel Multicriteria Model Applied to Cashew Chestnut Industrialization Process

Isabelle Tamanini, Ana Lisse Carvalho, Ana Karoline Castro,
and Plácido Rogério Pinheiro

Abstract. The industrialization process of cashew chestnut involves a decision making based on subjective criteria. It must be analyzed by the production manager, representing the decision maker, always aiming the choice of alternatives that maximize the entire almond index at the end of the process. Currently, this choice is carried out by the historical data verification, or considering the tacit experience of the manager. Therefore, the decision maker tends to miss the choice of the best possible solution. Due to the problem nature and to the necessity of the presentation of a good solution, the ZAPROS method was applied to the process. In the case study, the ZAPROS method application result was the relative and absolute ordinance of the alternatives, according to the DM's preferences. A case study was realized at the most critical part of the industrialization process of the cashew chestnut.

1 Introduction

One of the greatest problems faced by organizations is related to decision making process. The determination of the object which will conduct to the best result isn't a trivial process and involves series of factors to be analyzed. These problems are classified as complex and the consideration of all relevant aspects by the decision maker is practically impossible, due to human limitations.

The decision making related to management decisions is a critical process, since the wrong choice between two alternatives can lead to a waste of resources, affecting the company. Complex problems found in organizations can be solved in a valid and complete way through the application of multicriteria methods. This work was

Isabelle Tamanini · Ana Lisse Carvalho · Ana Karoline Castro · Plácido Rogério Pinheiro
Graduate Program in Applied Informatics, University of Fortaleza, Av. Washington Soares, 1321 - Bl J Sl 30 - 60.811-905 - Fortaleza - Brazil
e-mail: isabelle.tamanini@gmail.com, ana.carvalho@serpro.gov.br, akcastro@gmail.com, placido@unifor.br

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 243–252.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

developed due to the problem faced by the company Cascaju and based on previous studies realized there [9].

The chestnuts received in the companies present several physical characteristics (size, origin, harvest, humidity degrees) and are classified according to these characteristics. Then, the chestnuts go through series of mechanized industrial processes and, each one, influences on the classification. After that, the chestnuts are analyzed, chosen and stored, until they are submitted to the breaking process. The breaking process is composed by several stages, and this process presents the greatest difficulty for the decision making. Fig. 1 shows the cashew chestnut since it is taken from nature until it is ready to consume.

The problem is represented by the extraction of almonds of cashew chestnuts with a maximum amount of entire almonds. It represents a critical activity for the economic context of the northeast of Brazil.

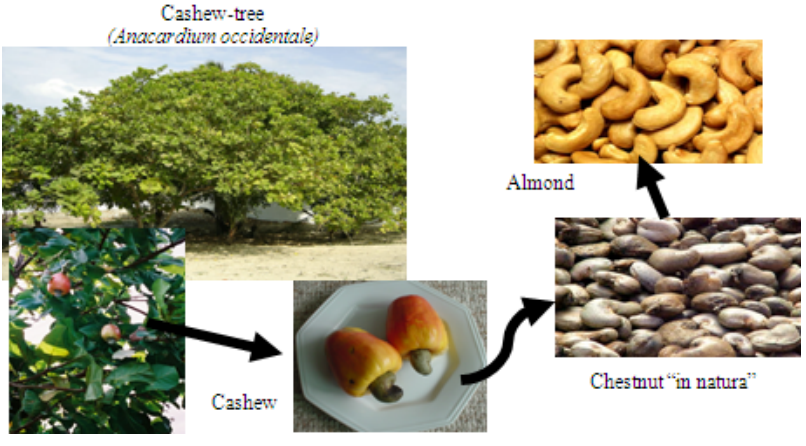


Fig. 1 Cashew Chestnut in nature and ready to consume

The reduction of the number of broken chestnuts in the industrialization process depends on the initial humidity. The chestnuts with certain degree of humidity before the decortication present minor breaking amount. The initial stage of the process is the humidification, that influences on the entire final almond number. For attainment of this ideal degree, the previous processes of decortication are adjusted in the production line. The adjustment takes into account values that can be used and defined as criteria and then commanded in levels of preferences by the decision maker. These values are related to the stages that precedes the decortication: humidification, baking, centrifugation and cooling, and involve values such as immersion and rest time in water, humidity rate, baking temperature, LCC viscosity, entrance outflow and cooling temperature. On the problem modeling based on ZAPROS method, these values were modeled as criteria and levels of preferences were established according to the preferences and experiences of the production manager, which represents the decision maker. The result of the modeling was a partial

order of the alternatives, taking into account the preferences considered ideal by the production manager.

A case study was carried out upon the afore mentioned stages aiming the maximization of the entire almond number. It was verified the applicability of ZAPROS method [6] and it was understood that other stages also can be shaped efficiently with this method of the Verbal Decision Analysis framework [2].

2 The Industrial Process of Breaking Cashew Chestnuts

The process is initiated when the cashew chestnut arrives from the supplier, and, then, a fast analysis of the raw material standards is carried out by the quality control. The raw material is bagged, and, posteriorly, stored.

After that, the chestnuts pass through a drying process. The drying is done using drying sheds, where the chestnuts are spread and dried by the heat of the sun refracted from a fiber glass roof. After the adequate drying, the raw material is automatically carried from the drying sheds to the classification area "in natura". In this stage the chestnuts are classified and stocked in accordance with their sizes (big, average, small, little and "cajui") and region of origin.

When removed from the stock, the cashew chestnuts pass to the washing stage, that aims cleaning superficially chestnuts and eliminating strange materials that are still present on the raw material. After that, the humidification process starts.

The humidification consists on humidifying the chestnut "in natura". For this stage, the chestnuts must be selected according to their classification (size, harvest, origin and initial humidity). The humidity is analyzed and it will influence on choosing the immersion and rest time. After the analysis, the chestnuts are placed in immersed silos with water (according to the immersion and rest time defined by the production manager). The humidity of the chestnut is modified during the humidification process. So, a new analysis of the humidity is necessary, since it influences on the definition of the LCC viscosity and on the entrance outflow of chestnuts in the baking recipients. Then, the chestnut is immersed in the LCC (Liquid of the Cashew Chestnut-peel) on a specific temperature and submitted to an average immersion time of 3 minutes.

The residual LCC present in the surface of the cashew chestnut-peel is removed and collected through a centrifugation process. After that, the chestnuts pass to the cooling process, on which they are cooled until they reach an acceptable level to the decortication process that follows. In this process, it is used a revolving equipment that, through a mechanical impact, allows opening the cashew chestnut-peel and removing the almond of its interior. Its velocity is electronically controlled (varying according to the current humidity degree of chestnuts) and the results are monitored in order to verify its efficiency.

Some successive processes to the decortication are:

- Separation: through special sieves and pneumatic separators, the almonds (entire ones and pieces) are separated of the peel. There are four sieves, each one with

a different velocity. When passing trough the first sieve, the chestnuts that weren't correctly separated from their peel are taken to the next sieve, and so on;

- Stewed: after the selection, the humidity of almonds must be reduced and the process of removing the peel that covers the almonds must be facilitated. So, the almonds are stewed on an average temperature of 70 degrees for an average time of 12 hours;

- Packing: the packing process uses a system of vacuum and carbonic gas injection, aiming keeping the integrity of the almond until its arrival to the final consumer.

After these processes, the product is stored in an appropriated way to keep physical integrity and, thus, to guarantee total quality to the consumer. The chestnut almonds are classified, for example, in the following products for sell: entire chestnut natural, granulated roast and flavoured chestnut.

3 Approaching the Critical Processes

Based on the decision maker's knowledge, the processes of humidification, baking, centrifugation and cooling were selected because they represented the greatest impact on the optimization of the entire almond number. Fig. 2 presents the relationship of these processes. The second column presents the critical stages of the process.

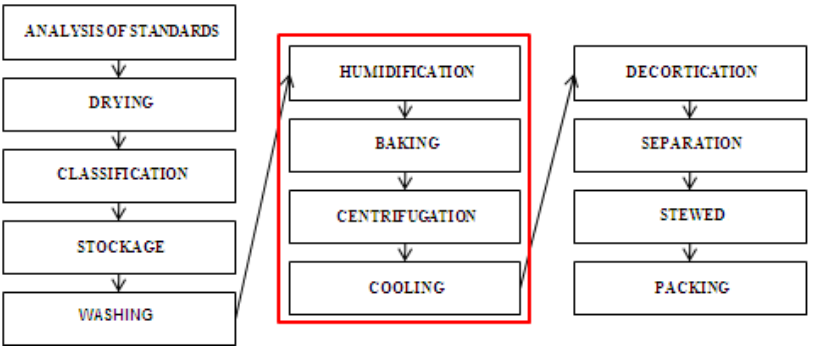


Fig. 2 Stages of the industrialization process of cashew chestnut

For each stage, the production manager makes decisions according to the necessary criteria that will lead to the best configuration to the whole process. This configuration is based on the preferential values of the decision maker, since it is defined by choices of purely qualitative values (immersion time, rest time, LCC viscosity, etc). Then, the values can be defined as criteria, transforming the decision making problem into an unstructured multicriteria problem [2]. In such case, the alternatives are identified based on the following questioning: how can we maximize the entire almonds number at the end of the industrialization process of cashew chestnut, considering the type of the chestnut, its harvesting, its origin and its humidity?

4 ZAPROS Method

The multicriteria method ZAPROS belongs to framework of Verbal Decision Analysis (VDA). It combines a group of methods that are essentially based on a verbal description of decision making problems. This method is structured on the acknowledgment that most of the decision making problems can be verbally described. The Verbal Decision Analysis supports the decision making process by verbal representation of the problem [5].

ZAPROS was developed with the aim of ranking given multicriteria alternatives, which makes it different from other verbal decision making, such as ORCLASS [7], and PACOM mainly due to its applicability. It uses the same procedures of elicitation of preferences, however with innovations related to the following aspects [6]:

- The procedure for construction of the ordinal scale for quality variation and criteria scales is simpler and more transparent;
- There is a new justification for the procedure of alternatives comparison, based on cognitive validations [4];
- The method offers both absolute and relative classification of alternatives.

The ZAPROS method can be applied to problems with the following characteristics [6]:

- The decision rule must be developed before the definition of alternatives;
- There is a large number of alternatives;
- Criteria evaluations for alternatives definition can only be established by human beings;
- The graduations of quality inherent to criteria are verbal definitions that represent the subjective values of the decision maker.

The decision maker is the key element of multicriteria problems and all necessary attention should be given in order to have well formed rules and consistent and correctly evaluated alternatives, considering the limits and capacity of natural language. Thus, the order of preference will be obtained according to the principle of good ordering established by Zorn's Lemma [1].

The application of the method ZAPROS for problem modeling can be applied following this three-step procedure:

1) Definition of the Criteria and Values of Criteria

Once the problem is defined, the criteria related to the decision making problem are elicited. The quality graduations of criteria are established through interviews and dialogues with specialists in the area and the decision makers. These should be presented on an ordered form, from the most preferable value to the less one.

2) Elicitation of Preferences In this stage, the scale of preferences for quality variations (Joint Scale of Quality Variations - JSQV) is constructed. The methodology follows the order of steps shown on fig. 3. This structure is the same proposed on [6], however, the substages 2 and 3 (numbered on the left side of the figure) were put together in just one substage.

The questions made considering the first reference situation are the same that the ones made considering the second reference situation. So, both situations will

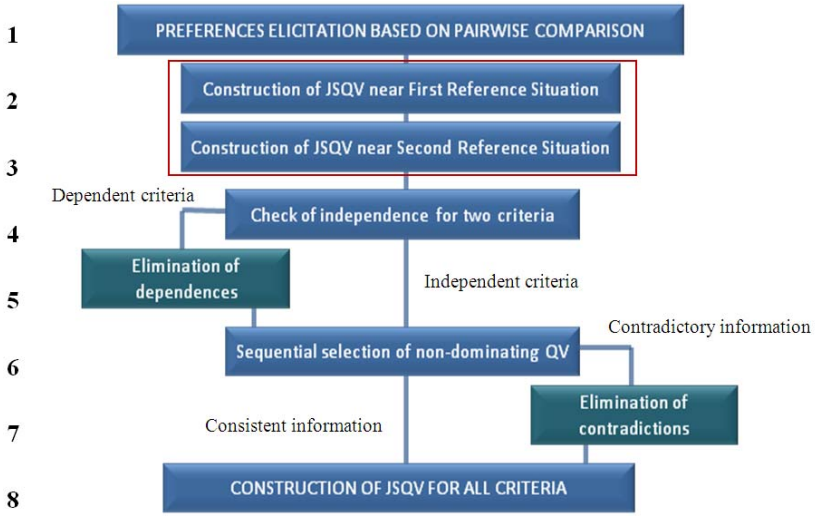


Fig. 3 Elicitation of preferences process

be presented and must be considered on the answer to the question, in order to not cause dependence of criteria. The questions to quality variations (QV) belonging to just one criteria will be made as follows: supposing a criterion A with three values $X_A = A_1, A_2, A_3$, the decision maker will be asked about his preferences between the QV $a_1 - a_2$, $a_1 - a_3$ and $a_2 - a_3$. Thus, there is a maximum of three questions to a criterion with three values $(n_q = 3)$.

The question will be formulated in a different way on the preferences elicitation for two criteria: it will be made dividing the QV in two items. For example, having the set of criteria k = A, B, C, where $n_q = 3$ and $X_q = q_1, q_2, q_3$. Considering the pair of criteria A, B and the QV a_1 and b_1, the decision maker should analyze which imaginary alternative would be preferable: A_1, B_2, C_1 or A_2, B_1, C_1. However, this answer must be the same to the alternatives A_1, B_2, C_3 and A_2, B_1, C_3. If the decision maker answers that the first option is better, then b_1 is preferable to a_1, because it is preferable to have the value B_2 on the alternative instead of A_2.

The transitivity also helps on checking the independence of criteria and groups of criteria as well as on the identification of contradictions on the decision maker's preferences. The dependence of criteria and contradictions should be eliminated by making new questions to the decision maker and remodeling the criteria.

3) Comparison of Alternatives With the aim of reducing the incomparability cases, we apply the same structure proposed on [6], but modifying the comparison of pairs of alternatives' substage according to the one proposed on [8].

Each alternative has a function of quality - V(y) [6], depending on the evaluations on criteria that it represents. On [8], it is proposed that the vectors of ranks of criteria values, which represent the function of quality, are rearranged in ascending order. Then, the values will be compared to the corresponding position of another

alternative's vector of values based on Pareto's dominance rule. Meanwhile, this procedure was modified to implementation because it was originally proposed to scales of preferences of criteria values, not for quality variations' scales.

So, supposing the comparison between alternatives $Alt1 = A_2, B_2, C_1$ and $Alt2 = A_3, B_1, C_2$, considering a scale of preferences: $a_1 \prec b_1 \prec c_1 \prec a_2 \prec b_2 \prec c_2 \prec a_3 \prec b_3 \prec c_3$, we have the following functions of quality: V(Alt1) = (0, 0, 2) and V(Alt2) = (0, 3, 4), which represents the ranks of, respectively, b_1 and c_1, a_2. Comparing the ranks presented, we can say that Alt1 is preferable to Alt2.

In order to establish the rank of alternatives, we have that an alternative is classified with rank i if it is dominated by the alternative of rank (i-1), an dominates one of rank (i+1) [8].

By these phases, a problem modeled based on the ZAPROS method results in an ordering of alternatives. The ordering gives us a quantitative notion of the order of preference, in an absolute way (in relation to all possible alternatives) as well as a relative way (in relation to a restricted group of alternatives).

5 Modeling the Industrialization Process of the Cashew Chestnut with ZAPROS

A tool developed for facilitating the application of the method ZAPROS [10] was applied to the modeled problem, aiming its solution and, consequently, the increase of the entire almonds number.

The criteria identified by means of interviews and dialogues with the decision maker, which represents the production manager of the company, were defined at the tool. The criteria are related to the processes of humidification, baking, centrifugation and cooling, and the values identified for them are exposed on table 1. The values for each criterion are ordered according to the DM's preferences, such that the most preferable value for a criterion is the first of the list (for example, A1).

In the next step, the Joint Scale of Quality Variation for the pair of dominant criteria is defined. According to the preferences of the DM and decisions conferences made, considering the criteria A (immersion time) and B (rest time), the variations of quality will be compared into an ideal situation, which is C1D1E1F1G1, and into the worst situation, C3D3E3F3G3 (criteria A and B are not included because they are being compared). This represents the comparison based on the best and the worst situations for the problem, and it must be considered on answering the questions in order to avoid inconsistences. Considering these situations and by means of questions to the manager, the Joint Scale of Quality Variation was elaborated for all the quality variations of criteria A and B $(a_1, a_2, a_3, b_1, b_2, b_3)$ as follows: $a_1 \prec b_1 \prec a_2 \prec b_2 \prec a_3 \prec b_3$.

The preferences will be elicited for all combinations of two criteria. The questions were answered by the DM, and the Joint Scale of Quality Variation for all criteria was defined as follows: $c_1 \prec a_1 \prec b_1 \prec d_1 \prec e_1 \prec f_1 \prec g_1 \prec c_2 \prec a_2 \prec b_2 \prec d_2 \prec e_2 \prec f_2 \prec g_2 \prec c_3 \prec a_3 \prec b_3 \prec d_3 \prec e_3 \prec f_3 \prec g_3$.

Table 1 Criteria involved on the cashew chestnut industrialization process

Criteria	Values of Criteria
A - Immersion Time	A1. 0 - 40 minutes
	A2. 41 - 80 minutes
	A3. 81 - 120 minutes
B - Rest Time in the Water	B1. 50 - 57 hours
	B2. 58 - 65 hours
	B3. 66 - 72 hours
C - Humidity degree	C1. 8,90 - 10,0
	C2. 10,1 - 12,4
	C3. 12,5 - 13,0
D - Baking Temperature	D1. 180 C - 198 C
	D2. 199 C - 216 C
	D3. 217 C - 235 C
E - LCC Viscosity	E1. 150 cps - 334 cps
	E2. 335 cps - 520 cps
	E3. 521 cps - 700 cps
F - Entrance Outflow	F1. 800 kg/h - 966 kg/h
	F2. 967 kg/h - 1.133 kg/h
	F3. 1.134 kg/h - 1.300 kg/h
G - Cooling Temperature	G1. 38 C - 45 C
	G2. 46 C - 53 C
	G3. 54 C - 60 C

Table 2 Alternatives classifications of cashew chestnut industrialization process

Alternatives	Evaluations on Criteria	Relative Rank	Absolute Rank
Alternative 1	A1B1C1D3E1F2G1	1	1
Alternative 2	A2B1C1D1E2F3G3	3	5
Alternative 3	A2B2C2D2E1F3G2	2	2
Alternative 4	A3B1C3D1E2F1G3	2	6
Alternative 5	A1B2C2D3E3F2G1	2	4
Alternative 6	A1B3C2D1E2F1G3	2	3

After having the Joint Scale of Quality Variation established for all the criteria, the alternatives were defined. The DM selected some alternatives among all the possible ones (all combinations of three values in seven criteria). Then, the comparison of alternatives process will be started in order to obtain the results for the problem. The rank generated at the first moment, corresponds to a relative classification of the alternatives (because not all possible alternatives were considered on the comparison).

If it's not possible to make a complete decision considering the obtained results due to incomparable alternatives, one can ask the comparison of all possible alternatives to the problem, which will obtain the absolute rank of each alternative

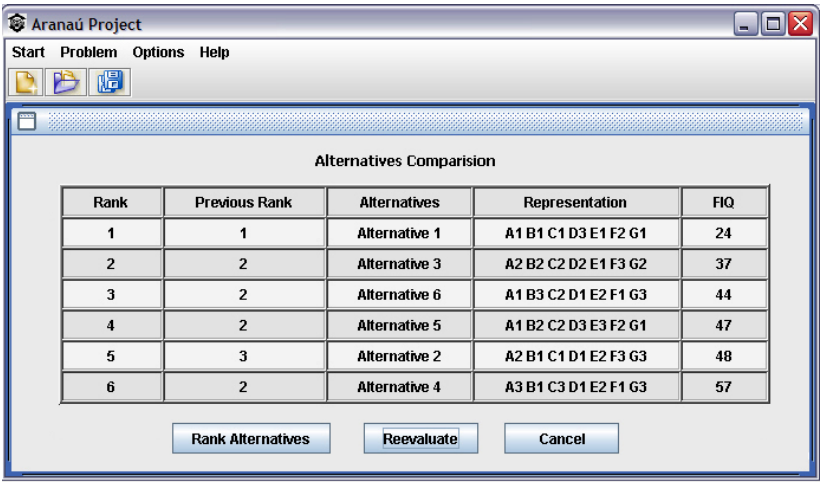


Fig. 4 Presentation of results

and establish an "absolute" rank for the considered alternatives. The alternatives and their criteria representations, the relative and the absolute ranks are exposed on table 2. Fig. 4 shows the presentation of results screen.

The order shown in table 2 represents a relative classification. Then, although alternative 1 (A1B1C1D3E1F2G1) is the best one of presented group of alternatives, it is not best among all the possible alternatives (the best alternative would be the ideal case "A1B1C1D1E1F1G1"). For the case study, the tool allows the insertion and analysis of new alternatives, in case of the DM needs to extend the presented solution. Therefore, the best decision is to adjust the values of the stages at the industrialization process to:

1. Immersion Time:0 - 40 minutes;
2. Rest Time in the Water: 50 - 57 hours;
3. Humidity degree after Humidify Process: 8,90 - 10, 0;
4. Baking Temperature: 217C - 235C;
5. LCC Viscosity: 150 cps - 334 cps;
6. Entrance Outflow: 967 kg/h - 1.133 kg/h;
7. Cooling Temperature: 38C - 45C.

6 Conclusions

The tool structured on the ZAPROS method of Verbal Decision Analysis was efficient on selecting the best alternative for the industrialization process of the cashew chestnut. The obtained classification allows the inclusion of new alternatives, without having to change the Joint Scale of Quality Variations. This represents great a

applicability for the exposed problem, although the amount combinations of values of the criteria is significant.

The optimization of the process in other stages (classification, decortication, separation and packing) of the industrialization process of the cashew chestnut are also necessary. Researches aiming the simplification of the qualitative values comparison are being carried out considering an hybridization of the ZAPROS method and neural networks. As future works, we also intend to structure new experiments on health areas will be done, aiming advances on early diagnosis of diseases.

Acknowledgements. The authors are thankful to CNPq for the support received on this project.

References

1. Halmos, P.R.: Naive Set Theory. Springer, New York (1974)
2. Figueira, J., Greco, S., Ehrgott, M. (eds.): Multiple Criteria Decision Analysis:State of the Art Surveys Series. International Series in Operations Research & Management Science, vol. 78. Springer, New York (2005)
3. Larichev, O.: Psychological validation of decision methods. Journal of Applied Systems Analysis 11(1), 37–46 (1984)
4. Larichev, O.: Cognitive validity in design of decision-aiding techniques. Journal of Multi-Criteria Decision Analysis 1(3), 127–138 (1992)
5. Larichev, O., Moshkovich, H.M.: Verbal decision analysis for unstructured problems. Kluwer Academic Publishers, Boston (1997)
6. Larichev, O.: Ranking Multicriteria Alternatives: The Method ZAPROS III. European Journal of Operational Research 131(3), 550–558 (2001)
7. Mechitov, A.I., Moshkovich, H.M., Olson, D.L.: Problems of decision rules elicitation in a classification task. Decision Support Systems 12, 115–126 (1994)
8. Moshkovich, H., Mechitov, A., Olson, D.: Ordinal Judgments in Multiattribute Decision Analysis. European Journal of Operational Research 137(3), 625–641 (2002)
9. Carvalho, A.L., de Castro, A.K.A., Pinheiro, P.R., Rodrigues, M.M., Gomes, L.F.A.M.: Model Multicriteria Applied to the Industrialization Process of the Cashew Chestnut. In: 3rd IEEE International Conference Service System and Service Management, pp. 878–882. IEEE Press, New York (2006)
10. Tamanini, I., Pinheiro, P.R.: Challenging the Incomparability Problem: An Approach Methodology Based on ZAPROS. In: An, L.T.H., Bouvry, P., Tao, P.D. (eds.) Modeling, Computation and Optimization in Information Systems and Management Sciences. Communications in Computer and Information Science, vol. 14, pp. 344–353. Springer, Heidelberg (2008)

Selection of Aggregation Operators with Decision Attitudes

Kevin Kam Fung Yuen

Abstract. The selection of the aggregation operators can be determined by the decision attitudes. This paper proposes an Aggregation Operators and Decision Attitudes (AODA) model to analyze the mapping relationship between aggregation operators and decision attitudes. A numerical example illustrates how decision attitudes of the aggregation operators can be determined by this model. This model can be applied in decision making systems with the selection problems of the appropriate aggregation operators with considerations of decision attitudes.

1 Introduction

Aggregation Operators (AO) are applied in many domains on the problems about the fusion of a collection of information granules. These domains include mathematics, physics, engineering, economics, management, and social sciences. Although the discussions of AO are very board, little research investigates the relationship with the decision attitudes. Yager [10] proposed valuation functions which used decision maker's attitudes and preferences as inputs to derive the final result with changing the parametric values. For this paper, as different aggregation operators produce different values, these results can be described by the possibility likelihoods for the decision attitudes. The selection of the aggregation operators is related to the likelihoods of the decision attitudes of the operators. To achieve the proposal, this paper proposes an Aggregation Operators and Decision Attitudes (AODA) model to analyze the mapping relationship between aggregation operators and decision attitudes on the basis of fuzzy set theory. The appropriate operators will be chosen according to the likelihood of the decision attitudes in AODA model.

2 Aggregation Operators

Definition 1: A generic aggregation operator A is a function which aggregates a set of granules $X = (x_1, \ldots, x_i, \ldots, x_n)$ into an Aggregated Value y. It has the form:

Kevin Kam Fung Yuen
The Hong Kong Polytechnic University
e-mail: kevinkf.yuen@gmail.com

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 255–264.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

$$y = A_{(n)}^{(t)}\left(\alpha;\left(x_1,\ldots,x_i,\ldots,x_n\right)\right) = A_{(n)}^{(t)}\left(\alpha;X\right)$$

t is the length of tuple(s) of x_i and n is the number of the granules. α is a construct parameter or a bag of construct parameters to scale A.

Sometimes, α is not shown if the information of α is not important for discussion in some scenarios. Likewise, AO can be simplified as the notations such as A, $A\left(\alpha;X\right)$, $A^{(t)}\left(\alpha;X\right)$ or $A_{(n)}^{(t)}\left(\alpha;X\right)$. This research is only interested in $t\in\left\{1,2\right\}$. To extend definition 1, then

Definition 2: A is non-weighted AO such that $x_i = c_i$ where c_i is single, or 1-tuple, and $c_i \in C$. Thus, $A^{(1)}\left(\alpha;X\right) = A^{(1)}\left(\alpha;\left(c_1,\ldots,c_i,\ldots,c_n\right)\right) = A^{(1)}\left(\alpha;C\right)$.

Alternatively, A is weighted AO such that $x_i = \left\{c_i, w_i\right\}$ where $w_i \in W = \left(w_1,\ldots,w_n\right)$ is the probability weight such that $\sum_{i\in\{1,\ldots,n\}} w_i = 1$. Thus x_i is a pair (or 2-tuple). Then,

$$A^{(2)}\left(\alpha;X\right) = A^{(2)}\left(\alpha;\left(\left\{c_1,w_1\right\},\ldots,\left\{c_i,w_i\right\},\ldots,\left\{c_n,w_n\right\}\right)\right).$$

Usually y and c_i have the fix interval $I' = \left[a,b\right] \subseteq \left[-\infty,\infty\right]$. $I = \left[0,1\right]$ is used in many studies. This is only a scaling matter between I' and I.

Definition 3: Let $I = \left[0,1\right]$. A non-weighted aggregation operator is the function $A : I^n \to I$. And a weighted aggregation operator is the function $A : W^T \times I^n \to I$. In both situations, $c_i, y \in I$.

3 Categories of Aggregation Operators

A Non-weighted AO is the special case of a weighted AO such that all weights are equal. This research focuses to discuss the weighted AO. Aggregation operators have been contributed by many researchers. Followings introduce the operators which are frequently used.

3.1 Quasi-Linear Means

The quasi-linear means [1, 6, 7] are of the general form: $qlm\left(W,C\right) = h^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\omega_i h\left(c_i\right)\right)$, $c \in I^p$. The function $h : I \to \Re$ is called the

generator of $qlm(w,c)$. If $h(x)=x^{\alpha}$, *qlm* is the weighted root power (*wrp*) or weighted generalized mean, and other three types are extended (table 1).

Table 1 Some forms of Quasi-linear means

1. Weighted root power $wrp(\alpha;W,C)=\left(\sum_{i=1}^{n} w_i {c_i}^{\alpha}\right)^{1/\alpha}$	2. Weighted harmonic mean ($\alpha \to -1$): $whm(W,C)=\dfrac{1}{\sum_{i=1}^{n} \frac{w_i}{c_i}}$
3. Weighted geometric mean ($\alpha \to 0$): $wgm(W,C)=\prod_{i=1}^{n} {c_i}^{w_i}$	4. Weighted arithmetic mean ($\alpha \to 1$) : $wam(W,C)=\sum_{i=1}^{n} w_i c_i$

3.2 Ordered Weighted Averaging

OWA [7, 8,11] is the weighted arithmetic mean which its weight values is related to the order position of C . $owa(W,C)=\sum_{i=1}^{n} w_i b_j$, where b_j is the *j*th largest of the C , $w_i \in [0,1]$ and $\sum_{i\in\{1,..,n\}} w_i = 1$. w_i can be generated from the form: $w_i = owaW = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right)$, where Q can be defined by $Q(\alpha;r)=r^{\alpha}$, $\alpha \geq 0$.

3.3 Weighted Median

In weighted median aggregation [7, 9], each element c_i is replaced by two elements: $c_i^{+}=(1-w_i)+w_i \cdot c_i$ and $c_i^{-}=w_i \cdot c_i$. Then the median value is computed by $wmed(W,C)=Median\left(c_1^{+},c_1^{-},\ldots,c_i^{+},c_i^{-},\ldots,c_n^{+},c_n^{-}\right)$. Alternatively, c_i^{+} and c_i^{-} can be computed by T-conorm and T-norm denoted as *S* and *T* respectively, and have the forms $c_i^{+}=S(1-w_i,c_i)$ and $c_i^{-}=T(w_i,c_i)$, which *S* and *T* are defined in following subsection.

3.4 T-Conorm and T-Norm

T-norms are with the properties which $T(x,1)=x$ and $T(x,y)\leq \min(x,y)$ while T-conorms have the properties which $S(x,0)=x$ and $S(x,y)\leq \max(x,y)$ [3]. Different kinds to T-norms and T-connorms [3, 7] are shown in table 2.

Table 2 various forms of T-norms and T-connorms

Min-Max	$Tm(a,b)=\min\{a,b\}$	$Sm(a,b)=\max\{a,b\}$
Lukasiewicz	$Tl(a,b)=\max\{a+b-1,0\}$	$Sl(a,b)=\min\{a+b,1\}$
Product/ Probabilistic	$Tp(a,b)=ab$	$Sp(a,b)=a+b-ab$
Dubois and Prade	$Tdp(\alpha;a,b)=\frac{a\cdot b}{\max\{a,b,\alpha\}}$, $\alpha\in(0,1)$	$Sdp(\alpha;a,b)=1-\frac{(1-a)(1-b)}{\max\{(1-a),(1-b),\alpha\}}$, $\alpha\in(0,1)$
Yager	$Ty(\alpha;a,b)=\max\left\{0,1-\left[(1-a)^{\alpha}+(1-b)^{\alpha}\right]^{1/\alpha}\right\}$	$Sy(\alpha;a,b)=\min\left\{1,\left(a^{\alpha}+b^{\alpha}\right)^{1/\alpha}\right\}$, $\alpha>0$
Frank	$Tf(\alpha;a,b)=\log_{\alpha}\left[1+\frac{\left(\alpha^{a}-1\right)\left(\alpha^{b}-1\right)}{\alpha-1}\right]$, $\alpha>0,\alpha\neq 1$	$Sf(\alpha;a,b)=1-\log_{\alpha}\left[1+\frac{\left(\alpha^{1-a}-1\right)\left(\alpha^{1-b}-1\right)}{\alpha-1}\right]$
Weber-Sugeno	$Tws(\alpha_T;a,b)=\max\left\{\frac{a+b-1+\alpha_T\cdot a\cdot b}{1+\alpha_T},0\right\}$, $\alpha_T>-1$	$Sws(\alpha_T;a,b)=\min\{a+b+\alpha_S\cdot a\cdot b,1\}$, $\alpha_S=\frac{\alpha_T}{(1+\alpha_T)}$
Schweizer & Sklar	$Tss(\alpha;a,b)=1-\left[(1-a)^{\alpha}+(1-b)^{\alpha}-(1-a)^{\alpha}(1-b)^{\alpha}\right]^{\frac{1}{\alpha}}$	$Sss(\alpha;a,b)=\left[a^{\alpha}+b^{\alpha}-a^{\alpha}b^{\alpha}\right]^{\frac{1}{\alpha}}$, $\alpha>0$

3.5 Weighted Gamma Operator

Zimmermann and Zysno [13] proposed a gamma operator on the unit interval based on T-norms and T-conorms. Calvo and Mesiar [2] modified the equation with weight assignment, which is of the form:

$$wgo(\alpha;C,W)=\left(\prod_{i=1}^{n}c_i^{w_i}\right)^{1-\alpha}\left(1-\prod_{i=1}^{n}(1-c_i)^{w_i}\right)^{\alpha}.$$

3.6 OWMAX and OWMIN

Ordered weighted maximum (*owmax*) and ordered weighted minimum operators (*owmin*) were proposed by [4]. Unlike OWA dealing with weighted arithmetic mean, owmax and owmin apply weighted maximum and minimum [6].

For any weight vector $W=(w_1,\ldots,w_n)\in[0,1]^n$, $1=w_1\geq\ldots\geq w_n$, *then* $\text{owmax}(W,C)=\bigvee_{i=1}^{n}(w_i\wedge c_{(i)})$, $C\in[0,1]^n$. For $W\in[0,1]^n$ $w_1\geq\ldots\geq w_n=0$, then $\text{owmin}(W,C)=\bigwedge_{i=1}^{n}(w_i\vee c_{(i)})$, $C\in[0,1]^n$.

3.7 Leximin Ordering

Leximin ordering was proposed by Dubois et al. [5]. Yager [10] improved a Lexmin ordering, based on OWA weights. Let Δ denote a distention threshold between the values being aggregated, Leximin is of the form: $\text{leximin}(W,C)=\sum_{i=1}^{n}w_ib_i$, where b_i is a sorted $C\in I^n$ in descending order such that $b_1>\ldots>b_n$. In addition,

$$w_j=LexW(\Delta,n)=\begin{cases}\dfrac{\Delta^{(n-j)}}{(1+\Delta)^{n-j}} & ,j=1\\ \dfrac{\Delta^{(n-j)}}{(1+\Delta)^{n+1-j}},j=2,\ldots,n\end{cases},\ w_j\in W$$

4 AODA Model

Under uncertainty, different people would have different decision attitudes. The decision attitudes (DAs) can be described by a collection of linguistic terms represented by a collection of DA fuzzy sets, i.e. $D=\{d_1,\cdots,d_j,\cdots,d_p\}$. Then,

Definition 4: A aggregated value y from an Aggregation Operator *A* belongs to a decision attitude fuzzy set d_j with the membership value $d_j(y)$ by the membership function $d_j:y\to I$.

The above definition implies the membership of a decision attitude fuzzy set is [0,1] from an aggregated value which also belongs to [0,1]. The membership is the likelihood of decision attitude in AODA model. As fuzzy set is characterized by this membership function, the same notation d_j is applied. Usually, the membership function applies triangular function $\mu_j(a,b,c)$ which is defined by three points.

Different input parameter sets, *X* 's, result in different Effective Aggregation Ranges (EAR) from a collection of the aggregation operators. The effective aggregation range $\left[y_*,y^*\right]$ is defined as follows:

Definition 5: Let the set of the aggregated values from the set $\tilde{A}$ of the aggregation operators $A's$ be $Y=(y_1,\ldots,y_k,\ldots,y_m)$. The permutation of Y is $\vec{Y}=\{y_{(1)},\ldots,y_{(k)},\cdots,y_{(m)}\}$, where $y_{(1)}\le y_{(2)}\cdots\le y_{(m)}$. Thus, the Effective Aggregation Range is $\left[y_*,y^*\right]$, where $y_*=y_{(1)}=\min(Y)$, $y^*=y_{(m)}=\max(Y)$.

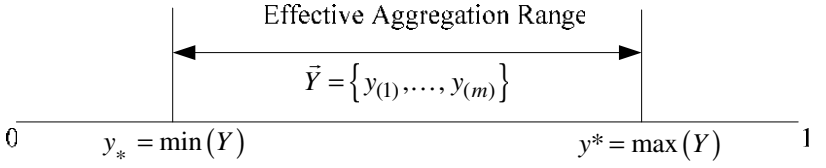


Fig. 1 Effective Aggregation Range

The EAR is the proper subset of *I*, *i.e.* $\left[y_*,y^*\right]\subseteq[0,1]$ (see Fig. 1). The collection of AO is of the form $\tilde{A}:X\to\left[y_*,y^*\right]^m$. Then a collection of decision attitudes fuzzy sets for an aggregation operator A is $D_A=\{\{y,d_1(y)\},\ldots,\{y,d_j(y)\},\ldots,\{y,d_p(y)\}\}$, where $y\in\left[y_*,y^*\right]$. Then a collection of DA fuzzy sets $D_{\tilde{A}}$ for a collection of aggregation operators $\tilde{A}=(A_1,\ldots,A_k,\ldots,A_m)$ is of the form:

$$D_{\tilde{A}}=\begin{pmatrix}\{\{y_{(1)},d_1\} & \ldots, & \{y_{(1)},d_j\} & \ldots, & \{y_{(1)},d_p\}\}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ \{\{y_{(k)},d_1\} & \ldots, & \{y_{(k)},d_j\} & \ldots, & \{y_{(k)},d_p\}\}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ \{\{y_{(m)},d_1\} & \ldots, & \{y_{(m)},d_j\} & \ldots, & \{y_{(m)},d_p\}\}\end{pmatrix},$$

where $\{y_{(k)},d_j\}=\{y_{(k)},d_j(y_{(k)})\},\forall k,\forall j$, and $y_{(1)}\le y_{(2)}\cdots\le y_{(m)}$ (Def. 5). To conclude, AODA algorithm is formed.

AODA Algorithm:

1. Define a collection of the membership functions of DA fuzzy sets $D=\{d_1,\cdots,d_j,\cdots,d_p\}$;
2. Define a collection of AO: $\tilde{A}=(A_1,\ldots,A_k,\ldots,A_m)$;
3. Get a collection of information granules: $X=(x_1,\ldots,x_i,\ldots,x_n)$;

4. Compute $\tilde{A}(X)$, and then $Y=(y_1,\ldots,y_k,\ldots,y_m)$ is achieved;
5. Get the permutation of *Y*: $\vec{Y}=\{y_{(1)},\ldots,y_{(k)},\cdots,y_{(m)}\}$;
6. Get $\left[y_*,y^*\right]$;
7. Assign intervals and memberships to *D*;
8. Calculate $D(\vec{Y})$;
9. Return $D_{\tilde{A}}$. **//END**

As this paper focus on discussion of the weighted aggregation operators, this follows $x_i=\{w_i,c_i\}\in X$ in step 3. To conclude, AODA is the function $g: X\to I$ and $g=\tilde{A}\circ D=D(\tilde{A}(X))$. This means that *g* maps the collection of information granules X with the set of the aggregators $\tilde{A}$, to the membership interval [0,1] corresponding to the collection of decision attitude fuzzy sets D. Usually the decision attitudes can be described by three linguistic terms: pessimistic, neutral and optimistic. Fig.2 shows some properties of the DA fuzzy sets.

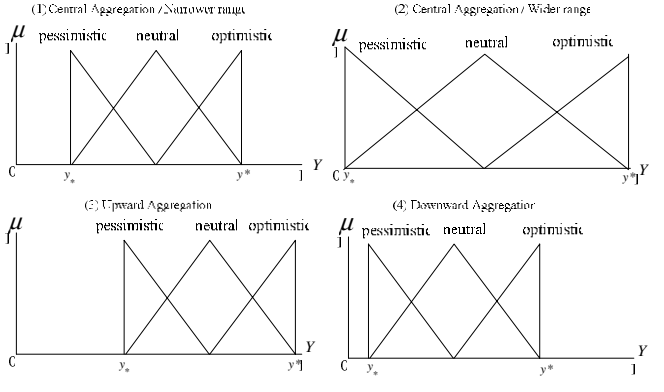


Fig. 2 Properties of EAR

The properties of EAR can be concluded as followings.

Proposition 1: Let $y'=mean(y_*,y^*)=\frac{1}{2}(y_*+y^*)$, and then

1. Effective aggregation range (EAR) is of downward aggregation if $y'<0.5$;.
2. EAR is of upward aggregation if $y'>0.5$;
3. EAR is of central aggregation if $y'=0.5$;
4. EAR 2 is more upward than EAR 1 if $y'_1<y'_2$.
5. EAR 2 is wider than EAR 1 if $y^*_1-y_{*_1}<y^*_2-y_{*_2}$.

Table 3 The results for $D_{\tilde{A}}$

(k)	A	$y_{(k)}$	$D\left(y_{(k)}\right)$
1	$wmed_f$	0.1193	{1,0,0}
2	owmax	0.3807	{0.0917,0.9083,0}
3	owmin	0.4	{0.0247,0.9753,0}
4	$wmed$	0.4619	{0,0.8095,0.1905}
5	$wmed_{ss}$	0.4868	{0,0.7231,0.2769}
6	$wmed_{mm}$	0.5	{0,0.6773,0.3227}
7	$wmed_{dp}$	0.5	{0,0.6773,0.3227}
8	$wmed_y$	0.5	{0,0.6773,0.3227}
9	wgo	0.5019	{0,0.6707,0.3293}
10	$wmed_l$	0.5127	{0,0.6333,0.3667}
11	whm	0.5137	{0,0.6298,0.3703}
12	$wmed_{ws}$	0.5199	{0,0.6080,0.3920}
13	wgm	0.5332	{0,0.5618,0.4382}
14	wrp	0.5375	{0,0.5470,0.4530}
15	wam	0.5557	{0,0.4838,0.5162}
16	owa	0.6949	{0,0,1}
17	Leximin	0.6949	{0,0,1}

5 A Numerical Example

From an AODA algorithm, a numerical example analysis is proposed as follows.

Step 1: Define the collection of decision attitude fuzzy sets:

Let $D=\{d_1,d_2,d_3\}$ represent the set of pessimistic, neural, and optimistic decision attitudes. $d_1=\mu\left(y_*,y_*,y'\right)$, $d_2=\mu\left(y_*,y',y^*\right)$, $d_2=\mu\left(y',y',y^*\right)$, where μ is the triangular membership function.

Step 2: Define a collection of the Aggregation Operators:

$$\tilde{A}=\left(A_1,\ldots,A_k,\ldots,A_{17}\right)=\begin{Bmatrix} wrp, whm, wgm, wam, owa, \text{owmax}, \text{owmin}, \\ \text{Lexmin}, wgo, wmed, wmed_l, wmed_{mm}, \\ wmed_{dp}, wmed_y, wmed_f, wmed_{ws}, wmed_{ss} \end{Bmatrix},$$

$wmed_l$ is *wmed* with Lukasiewicz T-norm and T-connorm. This naming convention is also applied to other *wmed* 's with different T-norms and T-connorms. In addition, if α affects the aggregation result, then a different value of α can be regarding as a different operator. Usually the best practice of α is suggested. In this example, $\alpha = 0.2$ is applied to all parametric operators.

Step 3: Get the collection of information granules:

Let $X = (x_1, \ldots, x_5)$ be weighted criteria such that $C = (0.4, 0.5, 0.6, 0.7, 0.9)$, and $W = owaW(0.6, 5) = (0.3801, 0.1964, 0.1589, 0.1387, 0.1253)$. Thus, $X = ((0.4, 0.1978), \ldots, (0.9, 0.6250))$.

Step 4: Compute Y by $\tilde{A}(X)$:

$$Y = \tilde{A}(X) = \begin{Bmatrix} 0.5375, 0.5137, 0.5332, 0.5557, 0.6949, 0.3807, 0.4, 0.6939, \\ 0.5019, 0.4619, 0.5127, 0.5, 0.5, 0.5, 0.1193, 0.5199, 0.4868 \end{Bmatrix}$$

Steps 5 and 6: Get the $\vec{Y}$, and $[y_*, y^*]$:

Get Ordering $(Y) = \{14, 11, 13, 15, 16, 2, 3, 16, 9, 4, 10, 6, 6, 6, 1, 12, 5\}$, then

$\vec{Y} = \begin{Bmatrix} 0.1193, 0.3807, 0.4, 0.4619, 0.4868, 0.5, 0.5, 0.5, 0.5019, 0.5127, \\ 0.5137, 0.5199, 0.5332, 0.5375, 0.5557, 0.6939, 0.6949 \end{Bmatrix}$. And

$[y_*, y^*] = [y_{(1)}, y_{(m)}] = [0.1193, 0.6949]$.

Step 7: Assign intervals and memberships to D.

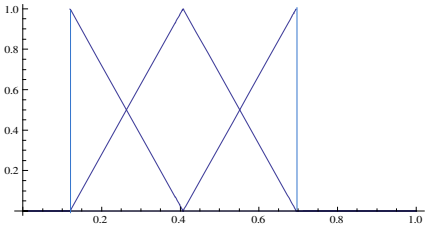


Fig. 3 AODA Pattern

Let $(y_*, y', y^*) = [0.1193, 0.4071, 0.6949]$ be substituted to $\mu(a, b, c)$ in D, and then AODA pattern is shown in Fig.3. It can be observed that the proposed numerical integration is downward integration since $y' = 0.4071 < 0.5$.

Steps 8 and 9: Calculate $D(\vec{Y})$ and return $D_{\tilde{A}}$. And the result is shown in Table 3.

6 Conclusions

This paper is devoted to a proposal how to map a collection of aggregation operators into a collection of the decision attitudes by the Aggregation Operators and Decision Attitudes (AODA) model which is useful when selecting suitable aggregation operators for the decision making applications with consideration of the decision attitudes of the decision makers. This research chooses the weighted aggregation operators with revision of 17 operators which are applied in AODA model. A numerical example is illustrated to validate the model. Future research will extend the AODA model and apply this model in some decision systems [e.g. 12].

References

[1] Bullen, P.S., Mitrinovic, D.S., Vasic, O.M.: Means and Their Inequalities. D. Reidel Publishing Company, Dordrecht (1988)
[2] Calvo, T., Mesiar, R.: Weighted triangular norms-based aggregation operators. Fuzzy Sets and Systems 137, 3–10 (2003)
[3] DetynieckI, M.: Mathematical Aggregation Operators and their Application to Video Querying. Thesis (2000)
[4] Dubois, D., Prade, H., Testemale, C.: Weighted fuzzy pattern-matching. Fuzzy Sets and Systems 28, 313–331 (1988)
[5] Dubois, D., Fargier, H., Prade, H.: Refinements of the maximin approach to decision-making in a fuzzy environment. Fuzzy Sets and Systems 81, 103–122 (1996)
[6] Marichal, J.L.: Aggregation operators for multicriteria decision aid. Ph.D. Thesis, University of Liège, Belgium (1998)
[7] Smolikava, R., Wachowiak, M.P.: Aggregation operators for selection problems. Fuzzy Sets and Systems 131, 23–34 (2002)
[8] Yager, R.R.: On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE trans. Systems, Man Cybernet. 18, 183–190 (1988)
[9] Yager, R.R.: On weighted median aggregation. Internat. J. Uncertainty, Fuzziness Knowledge-based Systems 2, 101–113 (1994)
[10] Yager, R.R.: On the analytic representation of Leximin ordering and its application to flexible constraint propagation. European J. Oper. Res. 102, 176–192 (1997)
[11] Yager, R.R.: OWA Aggregation over a Continuous Interval Argument With Applications to Decision Making. IEEE Trans. on Systems, Man and Cybernetics- Part B 34(5) (2004)
[12] Yuen, K.K.F., Lau, H.C.W.: A Linguistic Possibility-Probability Aggregation Model for Decision Analysis with Imperfect Knowledge. Applied Soft Computing 9, 575–589 (2009)
[13] Zimmermann, H.J., Zysno, P.: Latent connectives in human decision making. Fuzzy Sets and Systems 4, 37–51 (1980)

A New Approach Based on Artificial Neural Networks for High Order Bivariate Fuzzy Time Series

Erol Egrioglu, V. Rezan Uslu, Ufuk Yolcu, M.A. Basaran,
and Aladag C. Hakan

Abstract. When observations of time series are defined linguistically or do not follow the assumptions required for time series theory, the classical methods of time series analysis do not cope with fuzzy numbers and assumption violations. Therefore, forecasts are not reliable. [8], [9] gave a definition of fuzzy time series which have fuzzy observations and proposed a forecast method for it. In recent years, many researches about univariate fuzzy time series have been conducted. In [6], [5], [7], [4] and [10] bivariate fuzzy time series approaches have been proposed. In this study, a new method for high order bivariate fuzzy time series in which fuzzy relationships are determined by artificial neural networks (ANN) is proposed and the real data application of the proposed method is presented.

1 Introduction

As a matter of fact the observations of the most of the real life time series can be considered as uncertain observations. Consider the daily temperatures that vary during the day since the temperatures are measured at a certain moment of the day. So they imply the average of all temperatures measured during the day. This means that the observations of the temperature are not very exact and include uncertainty. This kind of uncertainty in measurements can be translated into fuzzy sets. Then it becomes an important issue to construct a model for a time series whose observations are fuzzy sets. Fuzzy set theory presented in [11] has a mathematical tool which helps for modeling this kind of uncertainty in many statistical methods. [8], [9] are the first studies that exploited fuzzy set theory. Song

Erol Egrioglu · V. Rezan Uslu · Ufuk Yolcu
Department of Statistics, Ondokuz Mayis University, Samsun 55139, Turkey
e-mail: {erole,rzzanu,yolcu}@omu.edu.tr

M.A. Basaran
Department of Mathematics, Nigde University, Nigde 51000, Turkey
e-mail: muratalper@yahoo.com

Aladag C. Hakan
Department of Statistics, Hacettepe University, Ankara 06800, Turkey
e-mail: chaldag@gmail.com.tr

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 265–273.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

and Chissom have made the definition of a fuzzy time series by using it and proposed a method that produces forecasts for fuzzy time series. [2] has improved the method of Song and Chissom and introduced an algorithm which uses a fuzzy logic group relation table for determining fuzzy relationships. [8], [9] and [2] are the studies based on a first order univariate model. The methods proposed in [3] and [1] are based on a high order univariate model. Since real life time series data are generally affected from many factors, a bivariate fuzzy time series model, instead of a univariate model, can give more accurate forecasting results. Because of the complexity of the real life data, bivariate (two factors) fuzzy time series approaches given in [6], [5], [7], [4] and [10] have been improved in order to obtain more comprehensive model. In this study, a new method used a high order bivariate (two-factors) fuzzy time series model, which fuzzy relationships are determined by ANN, is proposed and an application is presented.

The second section of the paper has presented the definitions of fuzzy time series. In Section 3 our proposed method is introduced. In Section 4 the proposed method is applied to the data of Taiwan Stock Exchange Capitalization Index Weighted Stock Index (TAIEX) and Taiwan Futures Exchange (TAIFEX). This section also contains the results of the proposed method and those of other methods available in the literature for the purpose of comparison. The final section is a detailed discussion of our results.

2 Fuzzy Time Series

The definition of fuzzy time series was firstly introduced by [8] and [9]. In contrast to conventional time series methods, various theoretical assumptions do not need to be checked in fuzzy time series approaches. The most important advantage of fuzzy time series approach is to be able to work with a very small set of data and not to require the linearity assumption. General definitions of fuzzy time series are given as follows:

Let U be the universe of discourse, where $U = \{u_1, u_2, ..., u_b\}$. A fuzzy set A_i of U is defined as $A_i = f_{A_i}(u_1)/u_1 + f_{A_i}(u_2)/u_2 + \cdots + f_{A_i}(u_b)/u_b$, where f_{A_i} is the membership function of the fuzzy set $A_i; f_{A_i} : U \to [0,1]$. u_a is a generic element of fuzzy set $A_i; f_{A_i}(u_a)$ is the degree of belongingness of u_a to $A_i; f_{A_i}(u_a) \in [0,1]$ and $1 \le a \le b$.

Definition 1. Fuzzy time series**.** Let $Y(t)(t = \ldots, 0, 1, 2, \ldots)$, a subset of real numbers, be the universe of discourse on which fuzzy sets $f_j(t)$ are defined. If $F(t)$ is a collection of $f_1(t), f_2(t), ...$ then $F(t)$ is called a fuzzy time series defined on $Y(t)$.

Definition 2. Fuzzy time series relationships assume that $F(t)$ is caused only by $F(t-1)$, then the relationship can be expressed as: $F(t) = F(t-1) * R(t, t-1)$,

which is the fuzzy relationship between $F(t)$ and $F(t-1)$, where * represents as an operator. To sum up, let $F(t-1)=A_i$ and $F(t)=A_j$. The fuzzy logical relationship between $F(t)$ and $F(t-1)$ can be denoted as $A_i \rightarrow A_j$ where A_i refers to the left-hand side and A_j refers to the right-hand side of the fuzzy logical relationship. Furthermore, these fuzzy logical relationships can be grouped to establish different fuzzy relationships.

The high order fuzzy time series model proposed by [3] is given as follows:

Definition 3. Let $F(t)$ be a fuzzy time series. If $F(t)$ is caused by $F(t-1), F(t-2), \ldots,$ and $F(t-n)$, then this fuzzy logical relationship is represented by

$$F(t-n),...,F(t-2),F(t-1) \rightarrow F(t) \tag{1}$$

and it is called the nth order fuzzy time series forecasting model.

Bivariate fuzzy time series model defined by [10] is given as follows:

Definition 4. Let F and G be two fuzzy time series. Suppose that $F(t-1)=A_i$, $G(t-1)=B_k$ and $F(t)=A_j$. A bivariate fuzzy logical relationship is defined as A_i, $B_k \rightarrow A_j$, where A_i, B_k are referred to as the left hand side and A_j as the right hand side of the bivariate fuzzy logical relationship.

Therefore, first order bivariate fuzzy time series forecasting model is as follows:

$$F(t-1),G(t-1) \rightarrow F(t) \tag{2}$$

Definition 5. Let F and G be two fuzzy time series. If $F(t)$ is caused by $(F(t-1),G(t-1)),(F(t-2),G(t-2)),\ldots,(F(t-n),G(t-n))$ then this fuzzy logical relationship is represented by

$$(F(t-1),G(t-1)),(F(t-2),G(t-2)),\ldots,(F(t-n),G(t-n)) \rightarrow F(t) \tag{3}$$

And it is called the two- factors *n*th order fuzzy time series forecasting model, where $F(t)$ and $G(t)$ are called the main factor fuzzy time series and the second factor fuzzy time series, respectively $(t=....0,1,2,....)$.

3 The Proposed Method

For the aim of forecasting of a time series, the use of a multivariate fuzzy time series model, instead of the model with a univariate, can provide better forecasts since a real life time series data has a complex structure and is affected by many other factors. [10] present an algorithm which analyzes a first order bivariate fuzzy time series forecasting model. [6], [5], [7] and [4] have used some high order bivariate fuzzy time series forecasting models. However, the algorithms of the

analysis mentioned in these studies are complicated and required too many calculations. Therefore, they waste time. Regarding of the negative sides of these methods in the literature, we propose a new approach in this paper. In the proposed method, fuzzy relationships are determined by using feed forward ANN. The steps of the algorithm of the method we propose are given below.

Algorithm

Step 1. The universe of discourses and subintervals are defined for two time series. The min and max values of the time series are denoted by $D_{\min}$ and $D_{\max}$, respectively. Then two positive numbers D_1 and D_2 can be chosen in order to define the universe of discourse $U = [D_{\min} - D_1, D_{\max} + D_2]$.

Step 2. Define fuzzy sets based on the universe of discourses. Based on defined U and V universe of discourses and subintervals, fuzzy sets $A_1, A_2, ..., A_{k_1}$ and $B_1, B_2, ..., B_{k_2}$ are defined and given below for time series and residuals, respectively.

$$A_1 = a_{11}/u_1 + a_{12}/u_2 + ... + a_{1n_1}/u_{r_1} \qquad B_1 = b_{11}/v_1 + b_{12}/v_2 + ... + b_{1n_2}/v_{r_2}$$
$$A_2 = a_{11}/u_1 + a_{12}/u_2 + ... + a_{1n_1}/u_{r_1} \qquad B_2 = b_{11}/v_1 + b_{12}/v_2 + ... + b_{1n_2}/v_{r_2}$$
$$\vdots \qquad\qquad \vdots$$
$$A_{k_1} = a_{11}/u_1 + a_{12}/u_2 + ... + a_{1n_1}/u_{r_1} \qquad B_{k_2} = b_{11}/v_1 + b_{12}/v_2 + ... + b_{1n_2}/v_{r_2}$$

where a_{ij} is the degree of membership values of u_i and $a_{ij} \in [0,1]$ for $1 \le i \le k_1$ and $1 \le j \le r_1$. r_1 and k_1 are the numbers of subintervals and fuzzy sets, respectively. In similar way, where b_{ij} is the degree of membership values of v_j and $b_{ij} \in [0,1]$ for $1 \le i \le k_2$ and $1 \le j \le r_2$. r_2 and k_2 are the numbers of subintervals and fuzzy sets, respectively.

Step 3. Fuzzify the observations. Each crisp value is mapped into a fuzzy set where its membership degree has maximum value. Fuzzy main factor time series is demonstrated by *F(t),* fuzzy second factor time series is demonstrated by *G(t).*

Step 4. Establish fuzzy relationship. In order to establish fuzzy relationships, ANN can be employed. The data which is used for training consist of the main factor fuzzy time series lagged variables $F(t-1), F(t-2), ..., F(t-n)$ and the second factor fuzzy time series lagged variables $G(t-1), G(t-2), ..., G(t-n)$ which are taken as the inputs of network. The main factor fuzzy time series $F(t)$ is used for the output of network. Feed forward neural network is trained in terms of these inputs and output. The number of neurons in the input layer is taken as *2n* while the number of neurons in the hidden layer (NNHL) can be decided by trial and error method. It is obvious that the number of neurons in the output layer should be 1. The illustration of feed forward neural network used in establishing fuzzy

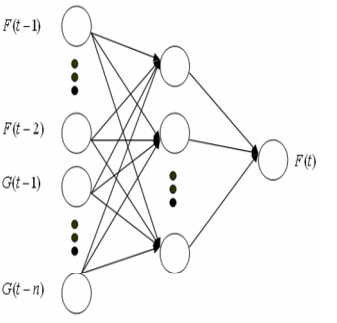


Fig. 1 Feed forward neural network architecture for fuzzy logic relation

relationship is shown Fig. 1. Detailed information about feed forward neural network can be found in [12] and [13].

Step 5. Forecast. Prepare data for forecasting:

$$F(t+k-1),...,F(t+k-n+1),F(t+k-n)\ ,G(t+k-1),...,G(t+k-n+1),G(t+k-n)$$

are taken as the inputs for the trained feed forward neural network and the output from the model is the fuzzy forecast for the $F(t+k)$.

Step 6. Defuzzify each fuzzy forecast $F(t+k)$. Apply "Centroid" method to get the results. This procedure (also called center of area, center of gravity) is the most often adopted method of defuzzyfication. Suppose that the fuzzy forecast of $F(t+k)$ is A_k. The defuzzyfied forecast is equal to the midpoint of the interval which corresponds to A_k.

4 Application

The method we propose has been applied to the time series data of Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) and Taiwan Futures Exchange (TAIFEX). In this application TAIFEX is the main factor time series while TAIEX is defined as the second factor time series. The application is given below step by step. First of all, the universe of discourses and subintervals for two time series are defined as

$D_{\min}=6251, D_{\max}=7599, D_1=51, D_2=1, U=[6200,7600]$ for TAIFEX and $D_{\min}=6200$, $D_{\max}=7560$, $D_1=0$ and $D_2=40$, $V=[6200,7600]$ for TAIEX.

The interval length for both time series is defined as 100 as in [7]. For this choice, the intervals are given below.

$u_1=[6200,6300], u_2=[6300,6400], u_3=[6400,6500]\ u_4=[6500,6600],$
$u_5=[6600,6700], u_6=[6700,6800], u_7=[6800,6900], u_8=[6900,7000],$

$u_9 = [7000,7100]$, $u_{10} = [7100,7200]$, $u_{11} = [7200,7300]$, $u_{12} = [7300,7400]$, $u_{13} = [7400,7500]$, $u_{14} = [7500,7600]$

$v_1 = [6200,6300]$, $v_2 = [6300,6400]$, $v_3 = [6400,6500]$, $v_4 = [6500,6600]$, $v_5 = [6600,6700]$, $v_6 = [6700,6800]$, $v_7 = [6800,6900]$, $v_8 = [6900,7000]$, $v_9 = [7000,7100]$, $v_{10} = [7100,7200]$, $v_{11} = [7200,7300]$, $v_{12} = [7300,7400]$, $v_{13} = [7400,7500]$, $v_{14} = [7500,7600]$

Based on defined U and V universe of discourses and subintervals, fuzzy sets $A_1, A_2, ..., A_{14}$ and $B_1, B_2, ..., B_{14}$ are defined and given below for two time series.

$$A_1 = a_{11}/u_1 + a_{12}/u_2 + ... + a_{1,14}/u_{14} \qquad B_1 = b_{11}/v_1 + b_{12}/v_2 + ... + b_{1,14}/v_{14}$$
$$A_2 = a_{21}/u_1 + a_{12}/u_2 + ... + a_{2,14}/u_{14} \qquad B_2 = b_{21}/v_1 + b_{22}/v_2 + ... + b_{2,14}/v_{14}$$
$$\vdots \qquad\qquad \vdots$$
$$A_{14} = a_{14,1}/u_1 + a_{14,2}/u_2 + ... + a_{14,14}/u_{14} \qquad B_{14} = b_{14,1}/v_1 + b_{14,2}/v_2 + ... + b_{14,14}/v_{14}$$

The observations are fuzzyfied. The fuzzyfied observations for the time series TAIFEX and TAIEX are demonstrated in Table 2. Feed forward ANN is used to determine fuzzy relationships. The inputs of the ANN are being changed according to the order of the model since it is changed 1 to 5 in the application. For example, the inputs of ANN for the second order model are $F(t-1), F(t-2), G(t-1), G(t-2)$ while the target value of ANN is $F(t)$. The outputs of ANN will be the predicted values of the fuzzy time series denoted by $\hat{F}(t)$. The number of neuron in the hidden layers of our ANN has been determined by changing the number of neurons from 1 to 4. The algorithm is applied for each of totally 20 cases that we determine. Then the calculated mean squared errors (MSE) for each case are listed in Table 1. As can be seen in Table 1, the best MSE value, 861.21, has been found for the case in which the model order is 5 and the number of neurons in the hidden layer is 4. The predicted values for this case are

Table 1 MSE values obtained from the proposed method

	Order of model				
Number of hidden layer neurons	**1**	**2**	**3**	**4**	**5**
1	11663.67	96276.64	9640.45	10224.09	9143.11
2	9027.80	6884.42	481350.5	4431.99	3535.73
3	8701.71	470742	4175.22	6718.50	1907.64
4	8432.15	470742	1704.08	1291.76	861.21

Table 2 Comparison of the forecasting values of TAIFEX and the mean square errors for different forecasting methods

Date	TAIEX	TAIFEX	Fuzzy TAIFEX	(Chen 1996)	(Huarg 2001) (two-variable heuristic)	(Huarng 2001) (three-variable heuristic)	(Lee et.al. 2006) two factors third order	The proposed Method (two factor-fifth order)
8.3.1998	7599	7552	14	-	-	-	-	-
8.4.1998	7593	7560	14	7450	7450	7450	-	-
8.5.1998	7500	7487	13	7450	7450	7450	-	-
8.6.1998	7472	7462	13	7500	7450	7500	7450	-
8.7.1998	7530	7515	14	7500	7500	7500	7550	-
8.10.1998	7372	7365	12	7450	7450	7450	7350	7350
8.11.1998	7384	7360	12	7300	7350	7300	7350	7350
8.12.1998	7352	7330	12	7300	7300	7300	7350	7350
8.13.1998	7363	7291	11	7300	7350	7300	7250	7250
8.14.1998	7348	7320	12	7183	7100	7188	7350	7350
8.15.1998	7372	7300	12	7300	7350	7300	7350	7350
8.17.1998	7274	7219	11	7300	7300	7300	7250	7250
8.18.1998	7182	7220	11	7183	7100	7100	7250	7250
8.19.1998	7293	7285	11	7183	7300	7300	7250	7250
8.20.1998	7271	7274	11	7183	7100	7188	7250	7250
8.21.1998	7213	7225	11	7183	7100	7100	7250	7250
8.24.1998	6958	6955	8	7183	7100	7100	6950	6950
8.25.1998	6908	6949	8	6850	6850	6850	6950	6950
8.26.1998	6814	6790	6	6850	6850	6850	6750	6750
8.27.1998	6813	6835	7	6775	6650	6775	6850	6850
8.28.1998	6724	6695	5	6850	6750	6750	6650	6650
8.29.1998	6736	6728	6	6750	6750	6750	6750	6750
8.31.1998	6550	6566	4	6775	6650	6650	6550	6550
9.1.1998	6335	6409	3	6450	6450	6450	6450	6450
9.2.1998	6472	6430	3	6450	6550	6550	6450	6450
9.3.1998	6251	6200	1	6450	6350	6350	6250	6250
9.4.1998	6463	6403	3	6450	6450	6450	6450	6450
9.5.1998	6756	6697	5	6450	6550	6550	6650	6650
9.7.1998	6801	6722	6	6750	6750	6750	6750	6750
9.8.1998	6942	6859	7	6775	6850	6850	6850	6850
9.9.1998	6895	6769	6	6850	6750	6750	6750	6750
9.10.1998	6804	6709	6	6775	6650	6650	6750	6750
9.11.1998	6842	6726	6	6775	6850	6775	6750	6750
9.14.1998	6860	6774	6	6775	6850	6775	6817	6750
9.15.1998	6858	6762	6	6775	6650	6775	6817	6750
9.16.1998	6973	6952	8	6775	6850	6850	6817	6950
9.17.1998	7001	6906	8	6850	6950	6850	6950	6950
9.18.1998	6962	6842	7	6850	6850	6850	6850	6850
9.19.1998	7150	7039	9	6850	6950	6850	7050	7050
9.21.1998	7029	6861	7	6850	6850	6850	6850	6850
9.22.1998	7034	6926	8	6850	6950	6850	6950	6950
9.23.1998	6962	6852	7	6850	6850	6850	6850	6850
9.24.1998	6980	6890	7	6850	6950	6850	6850	6850
9.25.1998	6980	6871	7	6850	6850	6850	6850	6850
9.28.1998	6911	6840	7	6850	6750	6750	6850	6850
9.29.1998	6885	6806	7	6850	6750	6850	6850	6850
9.30.1998	6834	6787	6	6850	6750	6750	6750	6750
	MSE			**9668**	**7856**	**5437**	**1364**	**861**

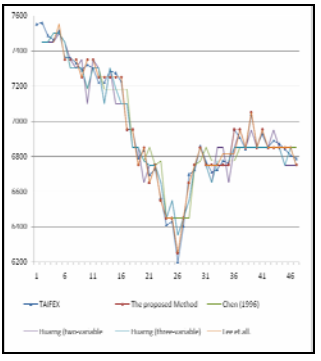


Fig. 2 The graph of TAIFEX time series together with the forecasts obtained by all the methods

found as shown in Table 2. Table 2 also presents the MSE values and the forecasted values obtained from the methods which are proposed in [2], [6] and [7]. When we examined the results we saw that the smallest MSE value has been obtained from the proposed method. The real time series, together with their forecasted values obtained by all the methods, are comparatively plotted in Fig. 2.

5 Conclusion

In this paper, we propose a new method based on ANN for analyzing a high order bivariate fuzzy time series forecasting model. After our proposed method and the other methods given in [3], [6] and [7] are applied to the time series data sets which are TAIFEX and TAIEX, it has been observed that the best forecasts are obtained from our proposed method which has the smallest MSE value. Using ANN, instead of determining fuzzy relation tables after intense calculations, makes the method more computationally easier. The method mentioned herein is superior than the other methods in terms of providing better forecasts and being practical.

References

1. Aladag, C.H., Basaran, M.A., Egrioglu, E., Yolcu, U., Uslu, V.R.: Forecasting in high order fuzzy time series by using neural networks to define fuzzy relations. Expert Systems with Applications 36, 4228–4231 (2009)
2. Chen, S.M.: Forecasting enrollments based on fuzzy time-series. Fuzzy Sets and Systems 81, 311–319 (1996)
3. Chen, S.M.: Forecasting Enrollments based on high-order fuzzy time series, Cybernetics and Systems. An International Journal 33, 1–16 (2002)
4. Cheng, C.H., Chen, T.-L., Huang, C.-C.: Fuzzy dual factor time series for stock index forecasting. Expert Systems with Applications (2007), doi:10.1016/j.eswa.2007.09.037
5. Hsu, Y.Y., Tse, S.M., Wu, B.: A new approach of bivariate fuzzy time series analysis to the forecasting of a stock index. International Journal of Uncertainity, Fuzziness and Knowledge-Based Systems 11(6), 671–690 (2003)

6. Huarng, K.: Heuristic models of fuzzy time series for forecasting. Fuzzy Sets and Systems 123(3), 369–386 (2001)
7. Lee, L.-W., Wang, L.-H., Chen, S.-M., Leu, Y.-H.: Handling forecasting problems based on two factors high order fuzzy time series. IEEE Transactions on Fuzzy Systems 14, 468–477 (2006)
8. Song, Q., Chissom, B.S.: Fuzzy time series and its models. Fuzzy Sets and Systems 54, 269–277 (1993a)
9. Song, Q., Chissom, B.S.: Forecasting enrollments with fuzzy time series- Part I. Fuzzy Sets and Systems 54, 1–10 (1993b)
10. Yu, T.K., Huarng, K.: A bivariate fuzzy time series model to forecast the TAIEX. Expert Systems with Applications 34(4), 2945–2952 (2008)
11. Zadeh, L.A.: Fuzzy Sets. Inform and Control 8, 338–353 (1965)
12. Zhang, G.P., Patuwo, B.E., Hu, Y.M.: Forecasting with Artificial Neural Networks: The State of the Art. International Journal of Forecasting 14, 35–62 (1998)
13. Zurada, J.M.: Introduction of Artificial Neural Systems. West Publishing, St. Paul (1992)

A Genetic Fuzzy System with Inconsistent Rule Removal and Decision Tree Initialization

Pietari Pulkkinen and Hannu Koivisto

Abstract. This paper presents a genetic fuzzy system for identification of Pareto-optimal Mamdani fuzzy models (FMs) for function estimation problems. The method simultaneously optimizes the parameters of fuzzy sets and selects rules and rule conditions. Selection of rules and rule conditions does not rely only on genetic operators, but it is aided by heuristic rule and rule conditions removal. Instead of initializing the population by commonly used Wang-Mendel algorithm, we propose a modification to decision tree initialization. Experimental results reveal that our FMs are more accurate and consist of less rules and rule conditions than the FMs obtained by two recently published genetic fuzzy systems [2, 3].

1 Introduction

One reason for the popularity of fuzzy models (FMs) is that they can be transparent if identified adequately. Recent trend has been the application of multiobjective evolutionary algorithms (MOEAs) to find FMs presenting trade-off between accuracy and interpretability criteria (see for example [2, 3, 11, 14, 16]). Those methods are also called genetic fuzzy systems (GFS) [4]. It was stated in [2] that most of MOEA based methods in literature consider only classification problems, that is, the output belongs to a set of pre-specified labels. Only few have covered function estimation problems, that is, the value of output is continuous (see for example

Pietari Pulkkinen
Department of Automation Science and Engineering, Tampere University of Technology, P.O. Box 692, FI-33101 Tampere, Finland
e-mail: pietari.pulkkinen@tut.fi

Hannu Koivisto
Department of Automation Science and Engineering, Tampere University of Technology, P.O. Box 692, FI-33101 Tampere, Finland
e-mail: hannu.koivisto@tut.fi

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 275–284.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

[2, 3, 14, 16]). Some of these [14, 16], however, applied Takagi-Sugeno (TS) FM, which may be harder to interpret than Mamdani FM due to linear function in its consequent part.

In [2, 3] MOEAs were applied to find a Pareto-optimal set of Mamdani FMs. In [2], the number of rules and parameters of fuzzy sets were optimized by MOEA, but the rule conditions were kept fixed. Thus, optimal rules can not be found. In [3], the number of rules and rule conditions were optimized by MOEA, however, the parameters of fuzzy sets were fixed and pre-specified. Therefore, fuzzy partitions may not present the real distribution of data and the accuracy of FMs is deteriorated [7]. On the other hand, it was pointed out in [8] that if the fuzzy partitions of each variable are available by a priori knowledge, modification of them can impair the interpretability of FMs.

When function estimation problems are considered, the initial population is usually obtained randomly or by Wang and Mendel (WM) method [15]. Naturally random initialization does not guarantee a good starting point for further optimization. WM method also has several drawbacks. First, it leads to high number of rules when high dimensional problems are considered. Second, it requires that the each variable is partitioned with fuzzy sets a priori. Third, it uses all available variables in all rules, thus leading to unnecessary complexity of the rule base.

DT based initialization algorithm was introduced in [1]. Because it can automatically select the relevant variables and partition the input space, it makes it a desirable initialization algorithm. In [1] it was applied to classification problems and here we modify it to suit for function estimation problems.

The further optimization is performed by MOEA, which selects the adequate rules and rule conditions and optimizes the parameters of fuzzy sets. Rule selection is not solely based on genetic operators but aided by heuristic removal of inconsistent rules and rule conditions. Our method is validated by identifying FMs for four well known problems and comparing our results to other MOEA based approaches [2, 3]. Our results show that our method obtains more compact and accurate FMs than in the comparative studies.

This paper is organized as follows. First, Mamdani FMs are defined. Then, in section 3, our identification method is introduced. After that, in section 4, the results comparison is performed. Finally, conclusions are given in section 5.

2 Mamdani Fuzzy Models

Let the dataset with D data points and n input variables be denoted as $\mathbf{Z} = [\mathbf{X}\ \mathbf{y}]$, where $\mathbf{X}$ is $D \times n$ input matrix and $\mathbf{y}$ is $D \times 1$ output vector. Mamdani fuzzy rules are denoted as:

$$R_i: \text{ If } x_1 \text{ is } A_{i,1} \ \ldots \text{ and } x_n \text{ is } A_{i,n} \text{ then } B_i,$$

where $A_{i,j}$, $j = 1,\ldots,n$, $i = 1,\ldots,R$, is an input membership function (MF), B_i is an output MF, and R is the number of rules. In order to reduce the computational costs, the output of FMs is computed here by approximation of centroid method [17, 3]:

$$\hat{y}_k = \frac{\sum_{i=1}^{R} \beta_i(\mathbf{x}_k)\bar{B}_i}{\sum_{i=1}^{R} \beta_i(\mathbf{x}_k)}, \tag{1}$$

where $\bar{B}_i$ is the center an output MF B_i, and β_i is the rule activation degree: $\beta_i(\mathbf{x}_k) = \prod_{j=1}^{n} A_{i,j}(x_{k,j})$. When output MFs are uniformly and symmetrically shaped, eq. (1) is an equivalent to centroid defuzzification [17].

3 Proposed Identification Method

3.1 Mamdani FM Initialization Using C4.5 Algorithm

In [1], C4.5 algorithm [12] was used to initialize fuzzy classifiers. Since we consider function estimation problems, the output data need to be discretized in order to use C4.5 algorithm. That is done by dividing the output to N_{out} crisp regions, where N_{out} is a positive integer. Each output value falls into one of these N_{out} regions and it is replaced with corresponding class label $S \in \{1,\ldots,N_{out}\}$, representing these regions. Then, C4.5 is used to match the input vectors to one of these class labels.

In Fig. 1 Box-Jenkins Gas Furnace data[1] is used as an example of applying C4.5 to predict continuous output. It is seen that crisp decision tree (DT) can not interpolate between rules. Thus, the resulting DT is transformed into a fuzzy classifier (FC) using the methodology given in [1]. However, the class labels in rule consequent are replaced with membership functions (MFs) in order to predict continuous outputs. To be able to cover the whole output range, the center of an output MFs need to be placed to lower χ_l and upper χ_u bounds of the range $\chi = \chi_u - \chi_l$. For the sake of interpretability, the MFs centers are evenly placed between lower and upper bounds:

$$c_1 = \chi_l \text{ and } c_j = c_{j-1} + \frac{\chi}{N_{out}-1}, j = 2,\ldots,N_{out}. \tag{2}$$

Center values of output MFs are one of those centers $\bar{B}_i = c_{S_i}$, where $S_i \in \{1,\ldots,N_{out}\}$, $i = 1,\ldots,R$. Since $\bar{B}_i$ is the only parameter needed in eq. (1) to compute the output of an FM, the type of an output MF can be, for example, triangular, Gaussian, singleton, or generalized bell (gbell) without affecting the result of computation. An example of an output partition with 5 Gbell MFs is presented in Fig. 1 (c). Gbell MFs are defined as:

$$\mu(x;a,b,c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}}, \tag{3}$$

where b defines the fuzziness of an MF. Like mentioned earlier, it has no effect on computation of the output of an FM and in Fig. 1 (c) it was set to 2.5. For the sake of interpretability, it is required that at the intersection points, the intersecting MFs have membership value of 0.5. Therefore, the distance from the center c_j of each MF to the intersection points is $a = \frac{\chi}{2(N_{out}-1)}$.

[1] That dataset is described later in section 4.1.

3.2 Merging of Fuzzy Sets

The FM initialized by C4.5 is likely to contain highly similar fuzzy sets and it is beneficial to merge them. As a similarity measure the following is used: $S(A_i,A_j) = \frac{|A_i \quad A_j|}{|A_i \quad A_j|}$, where and are the set theoretic intersection and union, respectively [13] . In this paper all pairs exceeding the user specified threshold Δ are merged. After merging, the resulting FM may have similar [13] or inconsistent rules in the rule base. Therefore it goes through heuristic removal of inconsistent rules and rule conditions described later in section 3.5. Then, the rest of the population is created by randomly replacing some parameters of the *simplified initial FM*, such that, the initial population is widely spread [11]. Finally, the rest of the population go through the heuristic rule and rule conditions removal and the initial population is ready for further optimization by MOEA performed by popular NSGA-II [6].

3.3 Coding of the Mamdani FM

Mamdani FM is presented with three real-coded vectors: antecedent vector $\mathbf{A}^*$, input MF parameter vector $\mathbf{P}$, and output MF parameter vector $\mathbf{S}^*$. Asterisks in $\mathbf{A}^*$ and $\mathbf{S}^*$ indicate that although real-coded values are optimized by MOEA, the values are rounded to the nearest integer when fitness evaluation is performed. The resulting integer vectors are respectively marked as $\mathbf{A}$ and $\mathbf{S}$.

$\mathbf{A}$ specifies for each rule $i = 1,\ldots,R$, which MF is used for variable $j = 1,\ldots,n_s$, where n_s is the number of variables used in simplified initial FM. Because C4.5 can select input variables, $n_s \leq n$:

$$\mathbf{A} = (A_{1,1},A_{1,2},\ldots,A_{1,n_s},\ldots,A_{R,1},A_{R,2},\ldots,A_{R,n_s}), \tag{4}$$

$A_{i,j} \in \{0,1,\ldots,M_j\}$, where M_j stands for the number of MFs in variable j in simplified initial FM. If $A_{i,j} = 0$, variable j is not used in rule i. Variable selection

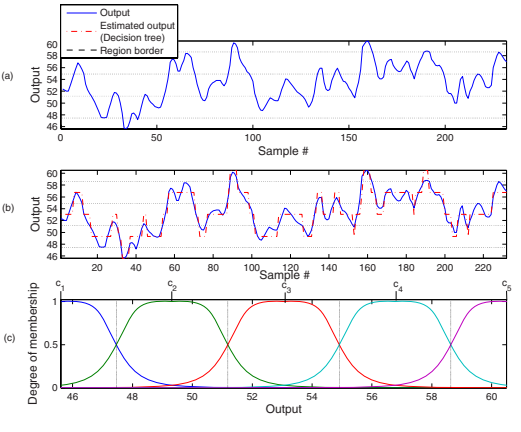


Fig. 1 (a) Each continuous output value belongs to one of the five regions and it is replaced with corresponding class label. (b) Estimating the output using crisp decision tree. (c) Each of the five regions is represented with an MF

and rule selection are possible during the evolutionary search. Variable j is removed if $\forall i, A_{i,j} = 0$ and rule i is removed if $\forall j, A_{i,j} = 0$.

P defines the parameters of input MFs:

$$\mathbf{P} = (P_{1,1}, P_{1,2}, \ldots, P_{1,\beta}, \ldots, P_{\gamma,1}, P_{\gamma,2}, \ldots, P_{\gamma,\beta}), \tag{5}$$

where γ stands for the number of parameters used to define an MF and $\beta = \sum_{j=1}^{n_s} M_j$ is the total number of MFs in simplified initial FM. Because Gbell MFs are applied in this paper, $\gamma = 3$.

The Gbell parameter a is constrained between $(0.005\chi, \chi/2)$, which means that the width of an MF should be at least 1% of the variable range χ and not wider than χ. Parameter b defining the fuzziness of an MF is required to be > 1 so that it does not cover large areas with low degree of membership, which leads to poor distinguishability of MFs [9]. On the other hand, large b value makes MFs almost crisp. Therefore $1 < b < 10$. Finally, the center of an MF is required to be inside the variable range. Thus, $\chi_l < c < \chi_u$.

During the evolutionary learning the antecedents of the rules and parameters of input MFs may be altered, such that, the initial output MFs may not be appropriate anymore [11]. Hence, $\mathbf{S} = (S_1, S_2, \ldots, S_R)$ stating the output MFs of the rules (see section 3.1) is optimized during the evolutionary search.

The total number of parameters to be optimized θ by MOEA is the sum of the lengths of $\mathbf{A}^*$, **P**, and $\mathbf{S}^*$. Thus, $\theta = R \times n_s + \gamma \times \beta + R$.

3.4 MOEA Optimization of the Initial Population

Each member of the population has the same amount of parameters to be optimized as in initial FM created by C4.5 and simplification operators. Fig. 2 depicts the further optimization of the initial population resulting to a set of Pareto-optimal FMs. Although C4.5 algorithm was used with discretized output data, during the further optimization the original undiscretized data are used. The fitness objectives to be minimized are: number of rules R, number of rule conditions (total rule length) R_{cond}, and mean squared error MSE $= \frac{1}{D}\sum_{k=1}^{D}(y_k - \hat{y}_k)^2$.

3.5 Heuristic Rule and Rule Condition Removal

At every iteration, NSGA-II evaluates the fitness of the offspring population [6]. The size of the offspring population is the same as the population size N_{pop}. Thus, the fitness of N_{pop} FMs is evaluated, which is a computationally demanding operation. If these FMs contain inconsistent rules and rule conditions, the fitness evaluations take a longer time than in case those inconsistencies are absent. Therefore the evolutionary search is aided by three heuristic operations: (**1**) Only one of the rules with exactly the same antecedent part is preserved [13]. (**2**) If there are rules of different length in which all conditions of the shorter rule(s) are present in the longer rule(s),

only one of them is preserved. By uniform chance, the preserved rule is either the longest rule (i.e. the most specific rule) or it is randomly selected out of the inconsistent rules [9]. (**3**) If there are conditions, which are present in all of the rules, they are removed from all of them [10].

4 Experiments

4.1 Datasets

The aim in *Electric* dataset is to estimate the maintenance costs of a medium voltage electrical network in a town [5] based on 4 input variables. It is a real-world dataset consisting of 1059 datapoints. Data partitions for 5-fold cross-validation were obtained from the website of Jorge Casillas: http://decsai.ugr.es/∼casillas/fmlib/.

Mackey-Glass (MG) is a chaotic time series defined by:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)}.$$

The problem is to predict $x(t+6)$ based on $[x(t), x(t-6), x(t-12)$, and $x(t-18)]$. In [3] and also in our experiments, τ was set to 17 and $x(0)$ to 1.2, which are typical values in the literature. Like in [3], we generated 500 datapoints to be used in our experiments.

Lorenz attractor (Lorenz) is another chaotic time series:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \sigma(y(t)-x(t)), \frac{\mathrm{d}y(t)}{\mathrm{d}t} = \rho x(t) - y(t) - x(t)z(t), \frac{\mathrm{d}z(t)}{\mathrm{d}t} = -z(t)\beta + x(t)y(t)$$

The standard values $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$ were used here and also in [3]. $x(0)$, $y(0)$, and $z(0)$ were set to 1. The goal is to predict $x(t+1)$ based on $[x(t), x(t-1), x(t-2), x(t-3)]$. Like in [3], 500 datapoints were generated to be used in our experiments.

Box-Jenkins Gas-Furnace (Gas) is a well known dataset. The problem is to predict CO_2 concentration $y(k)$ based on 4 previous concentration values $[y(k-1), \ldots, y(k-4)]$ and 6 previous methane feed rate values $[u(k-1), \ldots, y(k-6)]$. There are 296 datapoints in this dataset and it is available, for example, from the website of Greg Reinsel: http://www.stat.wisc.edu/∼reinsel/bjr-data.

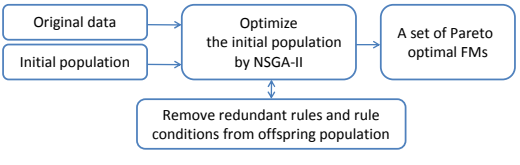


Fig. 2 Further optimization is performed using the original undiscretized data

4.2 Experimental Setup

For all the datasets, the parameters in Table 1 were applied. C4.5 was used with its default parameters defined in [12] and the output was partitioned with 5 output MFs in all experiments. The number of generations G and population size N_{pop} for Electric problem were both 1000 (1 000 000 fitness evaluations). For MG, Lorenz and Gas datasets $G = 350$ and $N_{\text{pop}} = 600$ (210 000 fitness evaluations). With all datasets, 5-fold cross-validation[2] was repeated 6 times (total 30 runs) with different random seeds.

Table 1 Parameters used in this study

Distribution index for mutation	Distribution index for cross-over	Cross-over probability	Mutation propability	Merging threshold Δ
20	20	0.9	$1/\theta$	0.7

The results are reported by drawing the Pareto fronts, which can be different for each run. There can also be FMs having the same number of rules but different number of rule conditions. To present an average result over a run, the average of the rule conditions R_{cond}, training accuracy MSE_{trn}, and testing accuracy MSE_{tst} of FMs with certain number of rules are computed for each run. Then those values are averaged over the runs they were present. To obtain reliable results, it is required that an FM with certain number of rules is present in the Pareto front at least in 15 out of 30 runs in order to be included in the Figures.

4.3 MG, Lorenz and Gas Datasets: Results Comparison

Our proposal is compared to a MOEA-based approach [3], which starts with random initialization of the population. Rules and rule conditions are selected but MFs are not tuned. As optimization algorithms, several well known algorithms, such as, NSGA-II, PAES, (2+2)PAES, and MOGA, were used. However, the results of MOGA are excluded, because the root mean squared error (RMSE) obtained by it was clearly worse than the RMSE of the other methods. Including those results would have made depicting of Figures, in which results are compared, difficult.

In that paper, a Table containing the information of the first FMs having RMSE for train set lower than a defined threshold was presented. The first FMs means that they are the most compact FMs of the Pareto front having the RMSE less than the threshold. Those thresholds were selected, such that, they present a good RMSE

[2] Each dataset was divided into 5 subsets. Learning was performed with 4 subsets and testing with the remaining subset. The procedure was iterated 5 times, such that, each time a different subset was used for testing.

value for a particular problem. For MG, Lorenz, and Gas datas the thresholds were 0.08, 1, and 0.6, respectively.

Our results, like mentioned in section 4.2, are the result of 5-fold cross-validation repeated 6 times. They are presented in Fig. 3. It is seen that for MG and Lorenz datasets, our FMs dominate the FMs of the comparative study. It is also noticed that for Gas data, none of our FMs is dominated by the FMs of the comparative study, which are dominated by many of our FMs.

4.4 Electric Data: Results Comparison

Our results were compared to another MOEA based Mamdani FM identification approach [2]. It creates the initial population using WM algorithm and uses MOEAs (NSGA-II [6] and SPEA2 [18]) to tune the MFs and to select rules (50 000 fitness evaluations). Note, that rule conditions are not selected by it but the whole rule is either included or not included into rule base. Thus each rule has as many rule conditions as there are input variables in the dataset.

The accuracy of the FMs was analyzed in [2] as follows. For each run, the FM with the lowest MSE_{trn} was selected and the number of rules R, number of rule conditions R_{cond} and MSE_{tst} of that FM were recorded. They were averaged over the number of runs and standard deviations σ of MSE_{trn} and MSE_{tst} were reported. Moreover, T-test with 95% confidence was reported for the MSE_{trn} and MSE_{tst}.

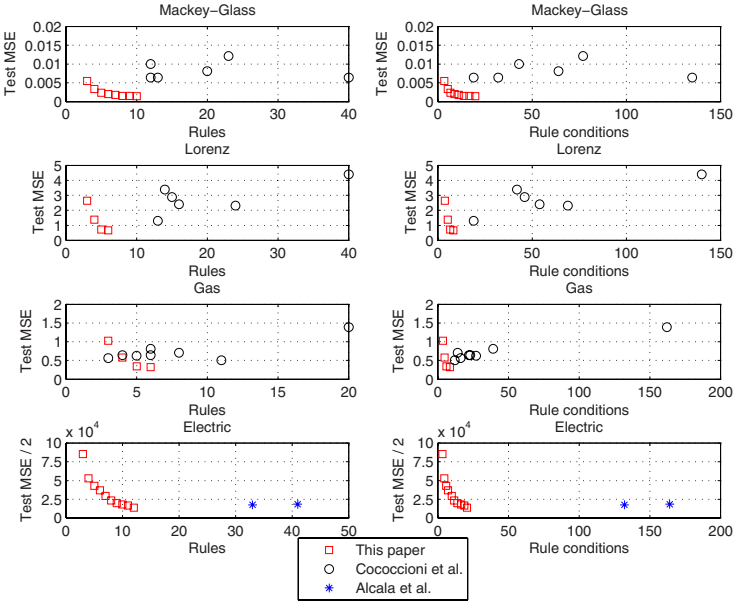


Fig. 3 The results comparison for four datasets

Here we perform exactly the same analysis and use the same notations as in [2]; $\star$ stands for the best averaged result in the column, $+$ means that the performance of the corresponding row is worse than the best result, and $=$ means that there is no significant difference compared to the best result.

From the results presented in Table 2 it is seen that the simplified initial FMs, denoted by C4.5, are more compact than the final FMs in comparative study. However, their accuracy need to be improved by learning process. It is seen that for both train and test sets, the T-test shows that our final FMs are significantly more accurate than final FMs in [2]. Yet, they contain far less rules and rule conditions than in comparative study. That can also be seen from Fig. 3, in which the Pareto front alongside with the results of [2] is shown.

Table 2 Electric data: results comparison

Reference	Method	R	R_{cond}	$\text{MSE}_{\text{trn}}/2$	σ_{trn}	T-test	$\text{MSE}_{\text{tst}}/2$	σ_{tst}	T-test
Alcalá	WM+SPEA2	33	132	13272	1265	+	17533	3226	+
et al. [2]	WM+NSGA-II	41	164	14488	965	+	18419	3054	+
This	C4.5	28.8	66.2	103155	8952	+	111760	21203	+
paper	C4.5+NSGA-II	13.4	23.8	11086	1993	$\star$	13035	3734	$\star$

5 Conclusions

We proposed a genetic fuzzy system (GFS) for identification of Mamdani fuzzy models (FMs). It simultaneously optimizes the parameters of fuzzy sets and selects rules and rule conditions. Learning process is not solely based on genetic operators but is aided by inconsistent rule removal. Besides, the initial population is created based on a modification to decision tree (DT) initialization proposed in [1].

Our method was compared to two genetic fuzzy systems [2, 3]. It was seen that usually our FMs contained less rules and rule conditions and were more accurate than the FMs of the comparative studies.

The focus of this paper was on obtaining accurate FMs with very small rule bases. Interpretability of fuzzy partition is not always guaranteed by our GFS and a future study is needed to consider that aspect as well.

References

1. Abonyi, J., Roubos, J.A., Szeifert, F.: Data-driven generation of compact, accurate, and linguistically-sound fuzzy classifiers based on a decision-tree initialization. Intl. J. Approx. Reason. 32(1), 1–21 (2003)
2. Alcalá, R., Alcalá-Fdez, J., Gacto, M.J., Herrera, F.: A multi-objective evolutionary algorithm for rule selection and tuning on fuzzy rule-based systems. In: IEEE Intl. Conf. Fuzzy Syst. (2007)

3. Cococcioni, M., Ducange, P., Lazzerini, B., Marcelloni, F.: A pareto-based multi-objective evolutionary approach to the identification of mamdani fuzzy systems. Soft Comput. 11(11), 1013–1031 (2007)
4. Cordón, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, L.: Ten years of genetic fuzzy systems: current framework and new trends. Fuzzy Set Syst. 141(1), 5–31 (2004)
5. Cordón, O., Herrera, F., Sánchez, L.: Solving electrical distribution problem using hybrid evolutionary data analysis techniques. Appl. Intelligence 10(1), 5–24 (1999)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6(2), 182–197 (2002)
7. Huang, H., Pasquier, M., Quek, C.: Optimally evolving irregular-shaped membership function for fuzzy systems. In: IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, pp. 11078–11085 (2006)
8. Ishibuchi, H., Nakashima, T.: Effect of rule weights in fuzzy rule-based classification systems. IEEE Trans. Fuzzy Syst. 9(4), 506–515 (2001)
9. Pulkkinen, P., Hytönen, J., Koivisto, H.: Detection of safe and harmful bioaerosols by means of fuzzy classifiers. In: IFAC World Congress, Seoul, Korea, pp. 12805–12812 (2008)
10. Pulkkinen, P., Koivisto, H.: Identification of interpretable and accurate fuzzy classifiers and function estimators with hybrid methods. Appl. Soft Comput. 7(2), 520–533 (2007)
11. Pulkkinen, P., Koivisto, H.: Fuzzy classifier identification using decision tree and multi-objective evolutionary algorithms. Intl. J. Approx. Reason. 48(2), 526–543 (2008)
12. Quinlan, J.R.: C4.5: Programs for Machine Learning (1993)
13. Setnes, M., Kaymak, U., van Nauta Lemke, H.R.: Similarity measures in fuzzy rule base simplification. IEEE Trans. SMC-B 28(3), 376–386 (1998)
14. Wang, H., Kwong, S., Jin, Y., Wei, W., Man, K.: Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction. Fuzzy Set Syst. 149(1), 149–186 (2005)
15. Wang, L.X., Mendel, J.M.: Generating fuzzy rules by learning from examples. IEEE Trans. on SMC 22(6), 1414–1427 (1992)
16. Xing, Z.Y., Zhang, Y., Hou, Y.L., Jia, L.M.: On generating fuzzy systems based on pareto multi-objective cooperative coevolutionary algorithm. Intl. J. Control, Automation, and Syst. 5(4), 444–455 (2007)
17. Zhang, B., Edmunds, J.: On fuzzy logic controllers. In: Intl. Conf. Control, Edinburgh, UK, pp. 961–965 (1991)
18. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm. In: Proceedings of the EUROGEN, pp. 19–26 (2001)

Robust Expectation Optimization Model Using the Possibility Measure for the Fuzzy Random Programming Problem

Takashi Hasuike and Hiroaki Ishii

Abstract. This paper considers an expectation optimization model using a possibility measure to the objective function in the fuzzy random programming problem, based on possibilistic programming and stochastic programming. The main fuzzy random programming problem is not a well-defined problem due to including random variables and fuzzy numbers. Therefore, in order to solve it analytically, a criterion for goal of objective function is set and the chance constraint are introduced. Then, considering decision maker's subjectivity and flexibility of the original plan, a fuzzy goal for each objective function is introduced. Furthermore, this paper considers that the occurrence probability of each scenario has ambiguity, and is represented as an interval value. Considering this interval of probability, a robust expectation optimization problem is proposed. Main problem is transformed into the deterministic equivalent linear programming problem, and so the analytical solution method extending previous solution approaches is constructed.

1 Introduction

In the real-world decision making problems, one often needs to make an optimal decision under uncertainty. Stochastic programming (for example, Beale [1], Charnes and Cooper [2], Dantzig [3]) and fuzzy programming (for example, Dubois and Prade [4], Inuiguchi and Tanino [5]) have been developed as useful tools for decision makers to determine an optimal solution. However, decision makers are faced with environments including both randomness derived from statistical

Takashi Hasuike
Graduate School of Information Science and Technology, Osaka University, Japan
e-mail: thasuike@ist.osaka-u.ac.jp

Hiroaki Ishii
Graduate School of Information Science and Technology, Osaka University, Japan
e-mail: ishii@ist.osaka-u.ac.jp

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 285–294.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

analysis based on practical data and fuzziness such as ambiguity of received information and decision maker's subjectivity. In order to construct a framework of decision making models under such stochastic and fuzzy environments, in some researches, fuzzy random variables (Kruse and Meyer [6], Kwakernaak [7], and Puri and Ralescu [8]) and random fuzzy variable (Liu [9, 10]) are brought to the attention of researchers.

Let us consider the situation that a profit per production unit is dependent on customer's demand, cost of each resource, etc., which are considered as future scenarios including random and ambiguous conditions. In the case that a realized profit under each scenario is estimated as random variables, a fuzzy set or fuzzy number, it turns out that the profit is expressed by a fuzzy random variable or a random fuzzy variable. In such a case, a linear programming problem is formulated to maximize total profit, in which the coefficients of objective function are fuzzy random variables or random fuzzy variables. Fuzzy random variables were first defined by Kwakernaak [7], and the mathematical basis was established by Puri and Ralescu [8]. Fuzzy random linear programming problems were investigated to provide decision making models and methodologies under fuzzy stochastic environments (For example, Katagiri et al. [11, 12]). Thus, the fuzzy random variable means a function from the probability space to the set of fuzzy numbers, i.e., a random variable taking fuzzy values.

In Katagiri et al. [13], they have considered an expectation optimization model using the possibility measure for fuzzy random programming problems. However, particularly in the case considering a finite random set to occurrence probabilities, it is difficult to set each probability as a fixed value due to uncertainty derived from a lack of reliable information and the subjectivity of the decision maker. Furthermore, in the case that decision makers consider the robustness of mathematical models in order to accommodate unpredictable situations as much as possible, it is natural that each probability has an ambiguity and is represented as an interval value. Therefore, in order to extend the previous model in Katagiri [13], and to develop a new versatile model for previous many fuzzy random programming problems by including intervals of occurrence probabilities, we propose a robust expectation optimization model for fuzzy random programming problems. By introducing such intervals to the probabilities and setting widths of the intervals and fuzzy numbers according to the subjective tendency of each decision maker, decision makers can consider not only single scenario case to assume initially but also various cases surrounding the single case, simultaneously.

In the mathematical programming, our proposed model is equivalently transformed into a semi-infinite programming problem. In general, it is difficult to solve this problem since we do not apply standard mathematical programming approaches, directly. However, using the properties of possibility measure and expectation, we show that our proposed model is efficiently solved by our proposed solution method.

This paper is organized as follows. The next section devotes to introducing fuzzy random variables and the formulation of the fuzzy random programming problem. In Section 3, we introduce a robust expectation optimization model for the main

fuzzy random programming problem, and construct the efficient solution method using the equivalent transformations and the duality of our proposed model. In Section 4, we illustrate a situation that can be applied by using a numerical example. Finally, in Section 5, we conclude this paper.

2 Formulation of Fuzzy Random Programming Problem

2.1 Fuzzy Random Variables

A fuzzy random variable was first defined by Kwakernaak [7]. The mathematical basis was established by Puri and Ralescu [8]. Kruse and Meyer [6] provides a slightly different definition. Since this article utilizes a simple one, we define fuzzy random variables as follows:

Definition 1. Let (Ω, B, P) be a probability space, $F(\mathbf{R})$ the set of fuzzy numbers with compact supports and X measurable mapping $\Omega \to F(\mathbf{R})$. Then, X is a fuzzy random variable if and only if given $\omega \in \Omega$, $X_\alpha(\omega)$ is a random interval for any $\alpha \in [0,1]$, where $X_\alpha(\omega)$ is a α-level set of the fuzzy set $X(\omega)$.

The above definition of fuzzy random variables corresponds to a special case of those given by Kwakernaak [7] and Puri and Ralesu [8]. The definitions of them are equivalent for the above case because a fuzzy number is a convex fuzzy set. Though it is a simple definition, it would be useful for various applications.

2.2 Fuzzy Random Programming Problem

In this paper, we consider the following linear programming problem:

$$\begin{aligned} &\text{Minimize } \sum_{j=1}^{n} \tilde{\bar{c}}_j x_j \\ &\text{subject to } \mathbf{A}x \le b, x \ge \mathbf{0} \end{aligned} \tag{1}$$

where x is an n-dimensional decision column vector, $\mathbf{A}$ is an $m \times n$ coefficient matrix and b is an m-dimensional column vector. Then, each $\tilde{\bar{c}}_j$ is a fuzzy random variable characterized by the following membership function:

$$\mu_{\tilde{\bar{c}}_j}(\omega) = \begin{cases} L\left(\frac{\bar{c}_j - \omega}{\alpha_j}\right) & (\bar{c}_j - \alpha_j \le \omega \le \bar{c}_j) \\ R\left(\frac{\omega - \bar{c}_j}{\beta_j}\right) & (\bar{c}_j \le \omega \le \bar{c}_j + \beta_j) \\ 0 & (\omega < \bar{c}_j - \alpha_j, \bar{c}_j + \beta_j < \omega) \end{cases} \tag{2}$$

where $L(x)$ and $R(x)$ are nonincreasing reference functions to satisfy $L(0) = R(0) = 1$, $L(1) = R(1) = 0$ and the parameters α_j and β_j represent the spreads corresponding to the left and the right sides, respectively, and both parameters are positive

values. Then, $\bar{c}_j$ is a random variable whose realization under the occurrence of the ith scenario is c_{ij}, and the probability that the ith scenario occurs denotes p_i. Suppose that $\sum_{i=1}^{S} p_i = 1$. Therefore, using this membership function and the extension principle, the membership function of the objective function $\tilde{\bar{Z}} = \sum_{j=1}^{n} \tilde{\bar{c}}_j x_j$ is as follows:

$$\mu_{\tilde{\bar{Z}}}(\omega) = \begin{cases} L\left(\dfrac{\sum_{j=1}^{n} \bar{c}_j x_j - \omega}{\sum_{j=1}^{n} \alpha_j x_j}\right) & \left(\sum_{j=1}^{n} \bar{c}_j x_j - \sum_{j=1}^{n} \alpha_j x_j \le \omega \le \sum_{j=1}^{n} \bar{c}_j x_j\right) \\ R\left(\dfrac{\omega - \sum_{j=1}^{n} \bar{c}_j x_j}{\sum_{j=1}^{n} \beta_j x_j}\right) & \left(\sum_{j=1}^{n} \bar{c}_j x_j \le \omega \le \sum_{j=1}^{n} \bar{c}_j x_j + \sum_{j=1}^{n} \beta_j x_j\right) \\ 0 & (otherwise) \end{cases} \tag{3}$$

In this paper, we call mathematical programming with fuzzy random variables *fuzzy random programming problem*. Problem (1) is not a well-defined problem due to including fuzziness and randomness, and it cannot be minimized in the sense of deterministic mathematical programming. Therefore, decision makers need to set a criterion with respect to the objective function and to transform the main problem into the deterministic equivalent problem. Furthermore, considering the ambiguity and subjectivity of the decision maker's judgment, it is natural to assume that the decision makers may have imprecise or fuzzy goals for the objective function in problem (1). In a minimization problem, a goal stated by the decision maker may be to achieve "approximately less than or equal to some value". This type of statement can be quantified by eliciting a corresponding membership function. For the objective function, we introduce a fuzzy goal characterized by the following membership function to consider the subjectivity of human's judgment:

$$\mu_{\tilde{G}}(y) = \begin{cases} 1 & (y \le g_L) \\ g(y) & (g_L < y \le g_U) \\ 0 & (g_U < y) \end{cases} \tag{4}$$

where $g(y)$ is a continuous and strictly decreasing function. By using a concept of possibility measure, the degree of possibility that the objective function value satisfies the fuzzy goal is represented as follows:

$$\Pi_{\tilde{\bar{Z}}}\left(\tilde{G}\right) = \sup_{y} \min\left\{\mu_{\tilde{\bar{Z}}}(y), \mu_{\tilde{G}}(y)\right\} \tag{5}$$

Using this degree of possibility, in the case that the decision maker prefers to maximize the degree of possibility, problem (1) is reformulated as follows:

$$\begin{aligned} &\text{Maximize } \Pi_{\tilde{\bar{Z}}}\left(\tilde{G}\right) \\ &\text{subject to } \mathbf{A}x \le b, x \ge \mathbf{0} \end{aligned} \tag{6}$$

This problem is equivalent to the following form introducing a parameter h:

$$\begin{aligned} &\text{Maximize } h \\ &\text{subject to } \Pi_{\tilde{Z}}^{\succeq}\left(\tilde{G}\right) \geq h, \\ &\qquad \mathbf{A}x \leq b, x \geq \mathbf{0} \end{aligned} \tag{7}$$

2.3 Expectation Optimization Model Using a Possibility Measure

Since $\Pi_{\tilde{Z}}^{\succeq}\left(\tilde{G}\right)$ varies randomly due to the randomness of $\tilde{c}_j$, problem (7) can be regarded as a stochastic programming problem. In stochastic programming problem, there are some typical models. In this paper, we focus on the expectation optimization model, which is to optimize the expectation of the objective function. In problem (7), since each scenario occurs at the probability p_i, the expected degree of possibility is expressed by

$$E\left[\Pi_{\tilde{Z}}^{\succeq}\left(\tilde{G}\right) \geq h\right] = \sum_{i=1}^{S} p_i \left(\sum_{j=1}^{n} c_{ij}x_j\right) - L^*(h) \sum_{j=1}^{n} \alpha_j x_j \leq g^{-1}(h) \tag{8}$$

where $L^*(h)$ is the pseudo inverse function of $L(x)$ and $g^{-1}(h)$ is the inverse function of $g(x)$. Therefore, problem (7) is reformulated the following expectation optimization model:

$$\begin{aligned} &\text{Maximize } h \\ &\text{subject to } E\left[\Pi_{\tilde{Z}}^{\succeq}\left(\tilde{G}\right)\right] \geq h, \\ &\qquad \mathbf{A}x \leq b, x \geq \mathbf{0} \end{aligned} \tag{9}$$

i.e.

$$\begin{aligned} &\text{Maximize } h \\ &\text{subject to } \sum_{i=1}^{S} p_i \left(\sum_{j=1}^{n} c_{ij}x_j\right) - L^*(h) \sum_{j=1}^{n} \alpha_j x_j \leq g^{-1}(h), \\ &\qquad \mathbf{A}x \leq b, x \geq \mathbf{0}, \sum_{i=1}^{S} p_i = 1 \end{aligned} \tag{10}$$

3 Robust Expectation Optimization Model

In problem (10), if each occurrence probability p_i is a fixed value, by using the bisection algorithm on h, problem (10) is equivalent to a linear programming problem, and so we obtain the optimal solution. However, a decision maker often does not assume each probability to be a fixed value due to uncertainty derived from a lack of reliable information and the subjectivity of the decision maker considering the robustness of the decision variables, but assumes them to include an interval to each probability. In this paper, this interval of probability is given as $p_i^L \leq p_i \leq p_i^U$, where the lower value p_i^L and upper value p_i^U are assumed to be constant. Thereby, it is possible that the decision maker constructs the flexible model

involving various practical conditions by considering the interval of probability. Then, in this subsection, we particularly focus on a maximin programming problem in which the decision maker considers the case that the objective function is less than a target value under all the situations including the interval of probabilities. This means that the proposed model is a robust model applied to various future cases such as more substantial changes of parameters. Therefore, we propose the following robust expectation optimization model:

$$\begin{aligned} &\text{Maximize } h \\ &\text{subject to } \max_{p\in P}\left\{\sum_{i=1}^{S} p_i\left(\sum_{j=1}^{n} c_{ij}x_j\right) - L^*(h)\sum_{j=1}^{n}\alpha_j x_j\right\} \le g^{-1}(h), \\ &\quad \mathbf{A}x \le b, x \ge \mathbf{0}, \\ &\quad P = \left\{ p \,\middle|\, p_i^L \le p_i \le p_i^U, \sum_{i=1}^{S} p_i = 1, (i = 1,2,...,S) \right\} \end{aligned} \tag{11}$$

i.e.

$$\begin{aligned} &\text{Maximize } h \\ &\text{subject to } \sum_{i=1}^{S} p_i\left(\sum_{j=1}^{n} c_{ij}x_j\right) - L^*(h)\sum_{j=1}^{n}\alpha_j x_j \le g^{-1}(h), (\forall p \in P) \\ &\quad \mathbf{A}x \le b, x \ge \mathbf{0}, \\ &\quad P = \left\{ p \,\middle|\, p_i^L \le p_i \le p_i^U, \sum_{i=1}^{S} p_i = 1, (i = 1,2,...,S) \right\} \end{aligned} \tag{12}$$

Since problem (12) is a semi-infinite programming problem (SIP) due to the inclusion of infinite inequalities $\sum_{i=1}^{S} p_i\left(\sum_{j=1}^{n} c_{ij}x_j\right) - L^*(h)\sum_{j=1}^{n}\alpha_j x_j \le g^{-1}(h), (\forall p \in P)$. Therefore, we need to construct the more efficient solution method as it cannot be solved directly using basic linear programming approaches. However, in the case that we fix the parameter p, problem (12) is equivalent to a finite linear programming problem. Furthermore, each decision variable x_j is a positive value. Therefore, we can apply an efficient solution method proposed by Lai and Wu [14] for SIP to problem (12). In order to construct the solution method, we introduce the following subproblem (SP(E)) where E is the index set defined by $E := \{1,2,...,L\}$:

$$\begin{aligned} &\text{Maximize } h \\ &\text{subject to } \sum_{i=1}^{S} p_i^{(l)}\left(\sum_{j=1}^{n} c_{ij}x_j\right) - L^*(h)\sum_{j=1}^{n}\alpha_j x_j \le g^{-1}(h), \left(\forall l \in E, p^{(l)} \in P\right) \\ &\quad \mathbf{A}x \le b, x \ge \mathbf{0}, \end{aligned} \tag{13}$$

In this case that we fix the parameter h as $\bar{h}$, we consider the following subproblem:

$$\begin{aligned} &\text{Maximize } \theta \\ &\text{subject to } \sum_{i=1}^{S} p_i^{(l)} \left(\sum_{j=1}^{n} c_{ij} x_j \right) - L^* \left(\bar{h} \right) \sum_{j=1}^{n} \alpha_j x_j \leq g^{-1} \left(\theta \right), \left(\forall l \in E, p^{(l)} \in P \right) \\ &\quad \mathbf{A}x \leq b, x \geq \mathbf{0}, \end{aligned} \tag{14}$$

Subsequently, let h^* be the optimal value of problem (13) and $\hat{h}$ denote h satisfying $\max_{p \in P} \left\{ \sum_{i=1}^{S} p_i \left(\sum_{j=1}^{n} c_{ij} x_j \right) - L^* (h) \sum_{j=1}^{n} \alpha_j x_j \right\} = g^{-1} (h)$. Then, the following theorem is derived from the study of Katagiri et al. [11].

Theorem 1. *Suppose that* $0 < h^* < 1$ *holds. Then,* h^* *is equivalent to be* $\hat{h}$.

In problem (14), the number of constraints in problem (14) is finite and all constraints are linear constraints, and so problem (14) is a standard linear programming problem. Furthermore, it is obvious that the dual problem DSP(E) for SP(E) is also a linear programming problem. Thereby, using SP(E) and DSP(E),we develop the following efficient solution algorithm for problem (11) extending the solution method of Lai and Wu [14].

Solution method

STEP1: Elicit the membership function of a fuzzy goal for the probability with respect to the objective function value.

STEP2: Set all intervals of probabilities $p_i^L \leq p_i \leq p_i^U$, $(i = 1, 2, ..., S)$.

STEP3: Set $U_q \leftarrow 1$ and $L_q \leftarrow 0$.

STEP4: Set $\gamma \leftarrow \frac{U_q + L_q}{2}$

STEP5: Set initial weight sets $E^0 = \left\{ 1, 2, ..., l^{(0)} \right\}$, $\left(\forall l \in E^0, p^{(l)} \in W \right)$ and solve problem (14) on $\bar{h} = \gamma$. Then, set the optimal solution to be x^0, θ^0, and $d \leftarrow 0$.

STEP6: If $Z_{\bar{h}} = \max_{p \in P} \left\{ \sum_{i=1}^{S} p_i \left(\sum_{j=1}^{n} c_{ij} x_j^d \right) - L^* \left(\bar{h} \right) \sum_{j=1}^{n} \alpha_j x_j^d - \theta^d \right\} \geq 0$, go to STEP9. If not, reset the values as follows:
$p^{l^d} := \left\{ p \left| \max_{p \in P} \left\{ \sum_{i=1}^{S} p_i \left(\sum_{j=1}^{n} c_{ij} x_j^d \right) - L^* \left(\bar{h} \right) \sum_{j=1}^{n} \alpha_j x_j^d - \theta^d \right\} \right. \right\}$ and $\bar{E}^{d+1} := E^d \cup \left\{ l^d \right\}$

STEP7: Solve the primal problem SP($\bar{E}^{d+1}$) and dual problem DSP($\bar{E}^{d+1}$), and obtain optimal solutions x^{d+1}, θ^{d+1} and $v_{d+1}(l)$

STEP8: Reset $E^{d+1} := \left\{ l \in \bar{E}^{d+1} \left| v_{d+1}(l) > 0 \right. \right\}$ and $d \leftarrow d + 1$. Then, return to STEP 6.

STEP9: If $Z_{\bar{h}} > g^{-1} \left(\bar{h} \right)$, $U_q \leftarrow \gamma$ and return to STEP4. If $Z_{\bar{h}} < g^{-1} \left(\bar{h} \right)$, $L_q \leftarrow \gamma$ and return to STEP4. Else, if $Z_{\bar{h}} = g^{-1} \left(\bar{h} \right)$, $x^{(d)}$ is an optimal solution of problem (11) and terminate this algorithm.

Subsequently, the solution method from steps 5 to 8 to solve SIP using primal problem SP(E) and dual problem DSP(E) is equal to that of Lai and Wu [14]. Therefore, it is obvious that convergence of this algorithm from steps 5 to 8 is obtained by

Lai and Wu [14]. Furthermore, the solution method form steps 1 to 4 and 9 is also equal to that of Katagiri [11]. Consequently, it is obvious that the convergence of this algorithm is entirely obtained.

4 Numerical Example

In order to illustrate our considering situation that the proposing solution method is applied, we provide a numerical example. We assume that there are four decision variables, three constraints and three scenarios to each fuzzy random variable set, i.e. $\bar{c} = \{c_1, c_2, c_3\}, p_k = \Pr\{\bar{c} = c_k\}$. The values of coefficient in objective functions are given as the following Tables 1. Then, all fuzzy numbers are assumed to be symmetric triangle fuzzy numbers.

Table 1 Values of parameters

Decision variable	A	B	C	D
$\tilde{\bar{c}}_j$	$\langle \bar{c}_1, 2\rangle$	$\langle \bar{c}_2, 5\rangle$	$\langle \bar{c}_3, 8\rangle$	$\langle \bar{c}_4, 3\rangle$
c_{1j}	5	10	15	15
c_{2j}	10	8	20	15
c_{3j}	8	12	15	10
p_1	[0.2, 0.4]	[0.2, 0.4]	[0.2, 0.4]	[0.2, 0.4]
p_2	[0.2, 0.5]	[0.2, 0.5]	[0.2, 0.5]	[0.2, 0.5]
p_3	[0.3, 0.6]	[0.3, 0.6]	[0.3, 0.6]	[0.3, 0.6]

Then, we consider the following problem as the numerical example:

$$
\begin{aligned}
&\text{Maximize } h \\
&\text{subject to } \max_{p\in P}\left\{\sum_{i=1}^{3} p_i\left(\sum_{j=1}^{4} c_{ij}x_j\right) - L^*(h)\sum_{j=1}^{n}\alpha_j x_j\right\} \le g^{-1}(h), \\
&\qquad 4x_1 + 4x_2 + 2x_3 + 5x_4 \ge 200, \\
&\qquad 3x_1 + 6x_2 + x_3 + 2x_4 \ge 250, \\
&\qquad 5x_1 + 3x_2 + 2x_3 + 3x_4 \ge 250, \\
&\qquad x_1, x_2, x_3, x_4 \ge 0
\end{aligned}
\tag{15}
$$

If the decision maker does not consider the fuzzy numbers, the main problem is transformed into the problem based on the stochastic programming whose object is minimizing $\max_{p\in P}\left\{\sum_{i=1}^{3} p_i\left(\sum_{j=1}^{4} c_{ij}x_j\right)\right\}$. We solve this problem and obtain the optimal value 502.38. Therefore, considering this optimal value, we assume that the aspiration level of objective value 500 becomes 0.5, and the fuzzy goal $\mu_{\tilde{G}}(y)$ in formula (4) is given as follows:

$$\mu_{\tilde{G}}(\omega) = \begin{cases} 1 & (\omega \leq 400) \\ \frac{600-\omega}{200} & (400 < \omega \leq 600) \\ 0 & (600 < \omega) \end{cases} \tag{16}$$

Using this membership function, we solve the problem using the proposed solution method in Section 3 and obtain the following optimal value.

Table 2 Optimal solution to our proposed problem

A	B	C	D	h
35.71	23.81	0.000	0.000	0.738

In this numerical example and the optimal solution, we find that the decision variables A and B with lower center values among all scenarios tend to be selected.

5 Conclusion

In this paper, we have considered a robust expectation maximization model for fuzzy random programming problems using the possibility measure to the objective function. In order to solve our proposed model analytically and efficiently, we have introduced the chance constraint and the fuzzy goal for the objective function, and performed the deterministic equivalent transformation to the main problem. Furthermore, by considering the intervals of occurrence probabilities, we have proposed the more versatile fuzzy random programming model than previous similar models.

It should be noted here that the problems involving fuzzy random variables are generally able to represent more complicate situations, but are a little complex problems compared with those involving only randomness or fuzziness. However, our proposed model can be solved not only easily but also optimally by combined use of conventional solution methods, i.e., simplex method and the bisection method. Furthermore, this model has the robustness to apply to various practical and changeable situations than the other fuzzy random programming models. As far as these points of view are concerned, our proposed model and solution algorithm have an advantage over other fuzzy random programming models.

References

1. Beale, E.M.L.: On optimizing a convex function subject to linear inequalities. Journal of the Royal Statistical Society 17, 173–184 (1955)
2. Charnes, A., Cooper, W.W.: Deterministic equivalents for optimizing and satisfing under chance constraints. Operations Research 11, 18–39 (1955)
3. Dantzig, G.B.: Linear programming under uncertainty. Management Science 1, 197–206 (1955)

4. Dubois, D., Prade, H.: Fuzzy Sets and Systems. Academic Press, New York (1980)
5. Inuiguchi, M., Tanino, T.: Portfolio selection under independent possibilistic information. Fuzzy Sets and Systems 115, 83–92 (2000)
6. Kruse, R., Meyer, K.D.: Statistics with Vague Data. D. Riedel Publishing Company (1987)
7. Kwakernaak, H.: Fuzzy random variable-1. Information Sciences 15, 1–29 (1978)
8. Puri, M.L., Ralescu, D.A.: Fuzzy random variables. Journal of Mathematical Analysis and Applications 14, 409–422 (1986)
9. Liu, B.: Theory and Practice of Uncertain Programming. Physica Verlag, Heidelberg (2002)
10. Liu, B.: Uncertainty theory. Physica Verlag, Heidelberg (2004)
11. Katagiri, H., Ishii, H., Sakawa, M.: On fuzzy random linear knapsack problems. Central European Journal of Operations Research 12(1), 59–70 (2004)
12. Katagiri, H., Sakawa, M., Ishii, H.: A study on fuzzy random portfolio selection problems using possibility and necessity measures. Scientiae Mathematicae Japonocae 65(2), 361–369 (2005)
13. Katagiri, H., Sakawa, M., Kato, K., Nishizaki, I.: Interactive multiobjective fuzzy random linear programming: Maximization of possibility and probability. European Journal of Operational Research 188, 530–539 (2008)
14. Lai, H.C., Wu, S.Y.: On linear semi-infite programming problems: an algorithm. Numerical Functional Analysis and Optimization 13, 287–304 (1992)

Improving Mining Fuzzy Rules with Artificial Immune Systems by Uniform Population

Edward Mężyk and Olgierd Unold

Abstract. The paper introduces speed boosting extension to a novel induction of fuzzy rules from raw data using Artificial Immune System methods. An improved approach uses a efficient initial population generation method. The so called uniform population method distributes initial population over chromosomes space uniformly omitting some drawbacks of random population. An improved mining method gives more stable results in an average shorter time.

1 Introduction

Data Mining is a part of a larger process called Knowledge Discovery in Databases. Its main goal is to find patterns within large data sets. Data mining tasks are often categorised into types of tasks they are applied to. One of them is a classification task, whose aim is to find general features of objects in order to predict classes they are associated with. Decision tree approach, classification rule learning, association rule mining, statistical approach, Bayesian network learning, and genetic algorithms are the most frequently used classification methods [5].

Fuzzy-based data mining is a modern and very promising approach to mine data in an efficient and comprehensible way. Moreover, fuzzy logic [10] can improve a classification task by using fuzzy sets to define overlapping class definitions. This kind of data mining algorithms discovers a set of rules of the form "IF (fuzzy conditions) THEN (class)", whose interpretation is as follows: IF an example's attribute

Edward Mężyk
Institute of Computer Engineering, Control and Robotics, Wroclaw University of Technology, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
e-mail: edward.mezyk@pwr.wroc.pl

Olgierd Unold
Institute of Computer Engineering, Control and Robotics, Wroclaw University of Technology, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
e-mail: olgierd.unold@pwr.wroc.pl

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 295–303.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

values satisfy the fuzzy conditions THEN the example belongs to the class predicted by the rule. The automated construction of fuzzy classification rules from data has been approached by different techniques like, e.g., neuro-fuzzy methods, genetic-algorithm based rule selection, and fuzzy clustering in combination with other methods such as fuzzy relations and genetic algorithm optimization (for references see [11]).

A quite novel approaches, among others, integrate Artificial Immune Systems (AISs) [3] and Fuzzy Systems to find not only accurate, but also linguistic interpretability fuzzy rules that predict the class of an example. The first AIS-based method for fuzzy rules mining was proposed in [2]. This approach, called IFRAIS (Induction of Fuzzy Rules with an Artificial Immune System), uses sequential covering and clonal selection to learn IF-THEN fuzzy rules. In [8] the speed of IFRAIS was improved significantly by buffering dicovered fuzzy rules in a clonal selection. One of the AIS-based algorithms for mining IF-THEN rules is based on extending the negative selection algorithm with a genetic algorithm [4]. Another one is mainly focused on the clonal selection and so-called a boosting mechanism to adapt the distribution of training instances in iterations [1]. A fuzzy AIS was proposed also in [9], however that work addresses not the task of classification, but the task of clustering.

This paper seeks to boost a speed of IFRAIS approach by exploring the use of initial uniform population.

This paper is organized as follows. Section 2 explains the details of the buffered IFRAIS. Section 3 introduces speed boosting extension to the IFRAIS by using uniform population. Section 4 briefly describes the used data set and discusses the experimental results. Finally section 5 concludes the paper with future works.

2 Buffered IFRAIS

Data preparation for learning in IFRAIS consists of the following steps: (1) create a fuzzy variable for each attribute in data set; (2) create class list for actual data set; (3) and compute information gain for each attribute in data set. IFRAIS uses a sequential covering as a main learning algorithm.

Sequential covering algorithm

```
Input: full training set
Output: fuzzy rules set

rules set = 0
FOR EACH class value c in class values list DO
  values count = number of c in full training set
  training set = full training set
  WHILE values count > number of maximal uncovered examples AND
     values count > percent of maximal uncovered examples
    rule = CLONAL SELECTION ALGORITHM(training set, c)
    covered  = COVER SET(training set, rule)
    training set = training set / covered with rule set
```

```
    values count = values count - size of covered
    ADD(rules set, rule)
  END WHILE
END FOR EACH
training set = full training set
FOR EACH rule R in rules set DO
  MAXIMIZE FITNESS(R, training set)
  COMPUTE FITNESS(R, training set)
END FOR EACH
RETURN rules set
```

In the first step a set of rules is initialized as an empty set. Next, for each class to be predicted the algorithm initializes the training set with all training examples and iteratively calls clonal selection procedure with the parameters: the current training set and the class to be predicted. The clonal selection procedure returns a discovered rule and next the learning algorithm adds the rule to the rule set and removes from the current training set the examples that have been correctly covered by the evolved rule. For all uncovered examples the most common class occured in a full training set is assigned.

Clonal selection algorithm is used to induct rule with the best fitness from training set. Basic elements of this method are antigens and antibodies which refers directly to biological immune systems. Antigen is an example from data set and antibody is a fuzzy rule.

Similarly to fuzzy rule structure, which consists of fuzzy conditions and class value, antibody comprises genes and informational gene. Number of genes in antibody is equal to number of attributes in data set. Each ith gene consists of a value v_i field involves one of the values belonging to the domain of the ith attribute in the rule condition, equal or not equal operator o_i, and an activation flag f_i that indicates whether ith fuzzy condition is active or inactive. The informational gene contains the value v_c of the class predicted by a rule. Each antibody can be represented as a set $\{(v_1,o_1,f_1),(v_2,o_2,f_2),\ldots,(v_n,o_n,f_n),v_c\}$, where n is the number of attributes in the rule condition. The antecedent of the rule is encoded as a variable-length list of rule conditions. If an attribute is not present in the antecedent of the rule, the value of its flag is 0.

Clonal selection algorithm in a buffered IFRAIS [8]

```
Input: training set, class value c
Output: fuzzy rule

Empty hash table BUFFER initiation
Randomly create antibodies population of size s
and class value c
FOR EACH antibody A in antibodies population
  PRUNE(A)
  IF CONTAINS(BUFFER, A)
    RETURN ASSOCIATED ELEMENT(BUFFER, A)
  ELSE
    fitness = COMPUTE FITNESS(A, training set)
```

```
    ADD(BUFFER, A, fitness)
END FOR EACH
FOR i=1 do generation count n
  WHILE clones populations size < s-1
    antibody to clone = TOURNAMENT SELECTION
        (antibody population)
    clones = CREATE x CLONES(antibody to clone)
    clones population =  clones population + clones
  END WHILE
  FOR EACH clone K in clones population
    muteRatio = MUTATION PROBABILITY(K)
    MUTATE(K, muteRatio)
    PRUNE(K)
    IF CONTAINS(BUFFER, K)
      RETURN ASSOCIATED ELEMENT(BUFFER, K)
    ELSE
      fitness = COMPUTE FITNESS(K, training set)
      ADD(BUFFER, A, fitness)
  END FOR EACH
Antibody population = SUCCESSION(antibody population,
clones population)
END FOR
result = BEST ANTIBODY(antibodies population)
RETURN result
```

In the first step a BUFFER table is initialized as an empty table. In a hash table BUFFER the pairs: rule and its fitness to actual training set are saved. If the hash table contains given rule, then fitness associated with this rule is returned as a result and an exhaustive computation of fitness is omitted. In other case rule fitness computation is executed and next rule with its fitness is saved in the hash table and computed value is returned as a function output.

Next, clonal selection generates randomly antibodies population with informational gene equals to class value c passed in algorithm parameter. Next each antibody from generated population is pruned. Rule pruning has a twofold motivation: reducing the overfitting of the rules to the data and improving the simplicity (comprehensibility) of the rules [12]. Fitness of the rule is computed according to the formula

$$FITNESS(rule) = \frac{TP}{TP+FN} \cdot \frac{TN}{TN+FP} \tag{1}$$

where TP is number of examples satisfying the rule and having the same class as predicted by the rule; FN is the number of examples that do not satisfy the rule but have the class predicted by the rule; TN is the number of examples that do not satisfy the rule and do not have the class predicted by the rule; and FP is the number of examples that satisfy the rule but do not have the class predicted by the rule. Hence the rules are fuzzy, the computation of the TP, FN, TN and FP involves measuring the degree of affinity between the example and the rule. This is computed by applying the standard aggregation fuzzy operator *min*

$$AFFINITY(rule, example) = min_{i=1}^{n}(\mu_i(att_i)) \quad (2)$$

where $\mu_i(att_i)$ denotes the degree to which the corresponding attribute value att_i of the example belongs to the fuzzy set accociated with the ith rule condition, and n is the number of the rule antecedent conditions. The degree of membership is not calculated for an inactive rule condition, and if the ith condition contains a negation operator, the membership function equals to $(1 - \mu_i(att_i))$ (complement). An example satisfies a rule if $AFFINITY(rule, example) > L$, where L is an activation threshold. For each antibody to be cloned the algorithm produces x clones. The value of x is proportional to the fitness of the antibody. Next, each of the clones undergoes a process of hypermutation, where the mutation rate is inversely proportional to the clone's fitness. Once a clone has undergone hypermutation, its corresponding rule antecedent is pruned by using the previously explained rule pruning procedure. Finally, the fitness of the clone is recomputed, using the current training set. In the last step the T-worst fitness antibodies in the current population are replaced by the T best-fitness clones out of all clones produced by the clonal selection procedure. Finally, the clonal selection procedure returns the best evolved rule, which will then be added to the set of discovered rules by the sequential covering. More details of the IFRAIS is to be found in [2].

3 Buffered IFRAIS with Uniform Population

Both standard IFRAIS[2] as well as a buffered IFRAIS [8] create initial population of fuzzy rules at random. Note that random initial population may be created in the infeasible region, or all the rules in population may be in the nearest neighborhood and far away from a solution, or search of solution may stick to a local solution and we can not get rid of this local solution. Local solution in the task of classification is the rules that have no agreeable generalization ability. In this study, creating initial population is performed in a systematic way inspired by the uniform population (UP) method [6].

Uniform population method distributes initial population over chromosomes space uniformly and then solution point has at least one chromosome of d neighborhood where d is a Hamming distance between solution point chromosome and the nearest chromosome to solution chromosome. The Hamming distance between two chromosomes is defined as the number of bit postions in which the two chromosomes differ.

UP works as follows. Initially, a chromosome is randomly generated and it is divided into r parts, where r is dividing factor and is determined by user or program. There are 2^r ways to take complement of one part or more than one part of r. So, it is possible to get $2^r - 1$ chromosomes from one randomly generated chromosome. When $r = 1$, then the inversion of a chromosome will be another individual in the initial population. Let $c = (x_1, x_2, \ldots, x_n)$ be a randomly created chromosome, and x_i is ith gene of c. Then $c_1 = (\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n})$ is derivated chromosome from c for $r = 1$.

If $r = 2$, then randomly created chromosome is divided into two equal parts. First, the inversion of the first part is taken and this yields another chromosome. Taking inversion of the second part will yield another chromosome, and inversion of all genes of randomly generated chromosome is also another chromosome. For $r = 2$ we derive from chromosome c three additional chromosomes $c_1 = (\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n})$, $c_2 = \left(\overline{x_1}, \overline{x_2}, \ldots, \overline{x_{\lfloor n/2 \rfloor}}, x_{\lfloor n/2 \rfloor + 1}, \ldots, x_n\right)$, $c_3 = \left(x_1, x_2, \ldots, x_{\lfloor n/2 \rfloor}, \overline{x_{\lfloor n/2 \rfloor + 1}}, \ldots, \overline{x_n}\right)$.

For $r = 3$ a chromosome is divided into three equal parts and seven new chromosomes are generated according to the above rules. This method can also be applied to all types of encoding such as binary encoding, floating point-based encoding, string-based encoding [6]. In real-type encoding, a random number can be generated in the half-open interval $[0, 1)$ and each gene complemented in the binary encoding are multiplied by this random number.

In the IFRAIS approach searched fuzzy rule is composed of n genes, where each gene corresponds to a condition containing one atrribute, and is partioned into three fileds: value v_i, equal or not equal operator o_i, and Boolean flag f_i. In such a case, complemented gene is derivated by negation of the operator o_i.

4 Experimental Results

In order to evaluate the performance of the speed boosting extensions, both a buffered IFRAIS [8] with random initial population and improved IFRAIS by uniform initial population were applied to six public domain data sets available from the UCI repository[1]. The experiments were conducted using a Distribution-Balanced Stratified Cross-Validation [13], which is a one of the version of well-known k-fold cross-validation, and improves the estimation quality by providing balanced intra-class distributions when partitioning a data set into multiple folds.

Table 1 shows the number of rows, attributes, continuous attributes, and classes for each data set. Note that only continuous attributes are fuzzified. The Votes data set does not have any continuous attribute to be fuzzified, whereas the other data sets have 6 or 9 continuous attributes that are fuzzified by IFRAIS. All experiments were repeated 50-times using 5-fold cross-validation on a 2.80 GHz Pentium with 2.00 GB RAM. Both versions of IFRAIS were implemented in C# language under .NET 3.5. Dividing factor r is equal to 3, initial population size of antibodies equals 50.

Table 2 shows for each data set the average accuracy rate and the average time of learning, both with standard deviations and difference between IFRAIS with random initial population (RIP) and IFRAIS with unified population (UP). As shown in Table 2 the IFRAIS improved by UP obtained comparable results as the standard version, but in considerably better time. For example, the time to achieve the same accuracy rate for Bupa set (ca 58.4 %) is 7% less for the improved IFRAIS (0.6859 ± 0.0177) than the standard one (0.7388 ± 0.0214 s). For the other data sets the time needed for learning for IFRAIS with UP is between 5 - 10% better. Only

[1] http://archive.ics.uci.edu/ml/datasets.html

Table 1 Data sets and number of rows, attributes, continuous attributes, and classes

Data set	#Rows	#Attrib.	#Cont.	#Class.
Bupa[1]	345	6	6	2
Crx[2]	653	15	6	2
Hepatitis	80	19	6	2
Ljubljana[3]	277	9	9	2
Wisconsin[4]	683	9	9	2
Votes[5]	232	16	0	2

[1] Bupa (Liver+Disorders).
[2] Crx (Credit+Approval).
[3] Ljubljana (Breast+Cancer).
[4] Wisconsin (Breast+Cancer+Wisconsin+(Original)).
[5] Votes (Congressional+Voting+Records).

Table 2 Accuracy rate and Learning Time on the test set for IFRAIS with random initial population (RIP) and IFRAIS with uniform population (UP). δ denotes difference between UP and RIP

Data set	Accuracy			Learning Time		
	RIP [%]	UP [%]	δ [%]	RIP [%]	UP [%]	δ [%]
Bupa	58.46±0.74	58.43±0.69	0.04	0.7388±0.0214	0.6859±0.0177	7.15
Crx	86.05±0.21	86.05±0.19	0.00	1.8702±0.0750	1.7733±0.0529	5.18
Hepatitis	77.08±1.53	77.85±1.66	-1.01	0.4786±0.0209	0.4363±0.0199	8.84
Ljubljana	69.29±1.19	69.20±1.25	0.13	0.8053±0.0213	0.7123±0.0159	11.55
Wisconsin	94.85±0.28	94.93±0.35	-0.08	1.6608±0.0793	1.6869±0.0618	-1.58
Votes	96.98±0.00	96.98±0.00	0.00	0.3335±0.0042	0.3369±0.0046	-1.01

for two data sets with best accuracy rates (Votes and Wisconsin) learning time is about 1% slower than for basic IFRAIS, but it may be considered as not significant because it is included between deviation range.

Table 3 shows the minimal and maximal accuracy rate of both versions of IFRAIS with difference between results achieved by those implementations. For most of the data sets values of minimal and maximal accuracy rate are higher for improved IFRAIS. It means higher probability that searched fuzzy rules receive higher quality. For example minimal accuracy for Bupa set (50,72%) is 6% better than this rate for basic implementation. So rules for Bupa set in worst case obtain accuracy rate higher than 50% what is very important while using IFRAIS without repetition. Because maximal accuracy for the same set didn't change, the range of accuracy

Table 3 Minimal i Maximal Accuracy rate on the test set for IFRAIS with random initial population (RIP) and IFRAIS with uniform population (UP). δ denotes difference between UP and RIP

Data set	Minimal Accuracy			Maximal Accuracy		
	RIP [%]	UP [%]	δ [%]	RIP [%]	UP [%]	δ [%]
Bupa	47.83	50.72	6.06	65.22	65.22	0.00
Crx	83.21	83.21	0.00	88.46	88.46	0.00
Hepatitis	62.50	62.50	0.00	93.75	100	6.67
Ljubljana	60.00	58.93	-1.79	78.18	80.00	2.33
Wisconsin	91.18	92.70	1.67	97.08	97.08	0.00
Votes	93.48	93.48	0.00	100	100	0.00

results became smaller, what is very desirable for IFRAIS used as fitness function. In such a case improved IFRAIS should give more stable results.

5 Conclusion

An AIS-based comprehensible fuzy rule mining that uses a efficient population generation method has been presented. With the proposed method the randomness in generating the initial population step in clonal selection has been removed and promising time results have been obtained.

It seems to be still possible to improve mining fuzzy rules with Artificial Immune Systems, considering not only time of working but also the effectiveness of the induced fuzzy rules. It could be achieved mostly by modifying the fitness function to reinforce the fitness of high-accuracy rules, as in [1]. We also consider changing the triangular membership functions to various more sophisticated functions and manipulating all system parameters to obtain higher quality results.

References

1. Alatas, B., Akin, E.: Mining fuzzy classification rules using an artificial immune system with boosting. In: Eder, J., Haav, H.-M., Kalja, A., Penjam, J. (eds.) ADBIS 2005. LNCS, vol. 3631, pp. 283–293. Springer, Heidelberg (2005)
2. Alves, R.T., Delgado, M.R., Lopes, H.S., Freitas, A.A.: An artificial immune system for fuzzy-rule induction in data mining. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 1011–1020. Springer, Heidelberg (2004)
3. Dasgupta, D. (ed.): Artificial Immune Systems and Their Applications. Springer, Heidelberg (1999)

4. Gonzales, F.A., Dasgupta, D.: An Immunogenetic Technique to Detect Anomalies in Network Traffic. In: Proceedings of Genetic and Evolutionary Computation, pp. 1081–1088. Morgan Kaufmann, San Mateo (2002)
5. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers/Academic Press (2001)
6. Karci, A., Arslan, A.: Uniform Population in Genetic Algorithms. Journal of Electrical and Electronics, Istanbul Unv. 2(2), 495–504 (2002)
7. Marsala, C.: Fuzzy Partitioning Methods, Granular Computing: An Emerging Paradigm, pp. 163–186. Physica-Verlag GmbH, Heidelberg (2001)
8. Mężyk, E., Unold, O.: Speed Boosting Induction of Fuzzy Rules with Artificial Immune Systems. In: Proc. of 12th WSEAS International Conference on SYSTEMS, Heraklion, Greece, pp. 704–706 (2008)
9. Nasaroui, O., Gonzales, F., Dasgupta, D.: The Fuzzy Artificial Immune System: Motivations, Basic Concepts, and Application to Clustering and Web Profiling. In: Proceedings of IEEE International Conference on Fuzzy Systems, pp. 711–716 (2002)
10. Pedrycz, W., Gomide, F.: An Introduction to Fuzzy Sets. Analysis and Design. MIT Press, Cambridge (1998)
11. Roubos, J.A., Setnes, M., Abonyi, J.: Learning Fuzzy Classification Rules from Labeled Data. Information Science 150, 77–93 (2003)
12. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Mateo (2005)
13. Zeng, X., Martinez, T.R.: Distribution-Balanced Stratified Cross-Validation for Accuracy Estimations. Journal of Experimental and Theoretical Artificial Intelligence 12(1), 1–12 (2000)

Incremental Locally Linear Fuzzy Classifier

Armin Eftekhari, Mojtaba Ahmadieh Khanesar, Mohamad Forouzanfar, and Mohammad Teshnehlab

Abstract. Optimizing the antecedent part of neuro-fuzzy system is investigated in a number of documents. Current approaches typically suffer from high computational complexity or lack of ability to extract knowledge from a given set of training data. In this paper, we introduce a novel incremental training algorithm for the class of neuro-fuzzy systems that are structured based on local linear classifiers. Linear discriminant analysis is utilized to transform the data into a space in which linear discriminancy of training samples is maximized. The neuro-fuzzy classifier is built in the transformed space, starting from the simplest form. In addition, rule consequent parameters are optimized using a local least square approach.

1 Introduction

Neuro-fuzzy system which benefits from both the computational power of neural network and logical power of fuzzy system is widely used in pattern recognition applications [1, 4, 13]. The performance of a neuro-fuzzy system is largely influenced by structure learning which involves two major issues: (i) parameter tuning of the antecedent part, which provides us with the fuzzy partitioning of the input space. (ii) parameter tuning of consequent part in which the parameters of consequent functions are obtained. Each subspace together with its associated consequent function is used to characterize a corresponding fuzzy rule. Generally, the local models (consequent functions) are chosen to be linear, which yields local linear model structures [2].

Recently, neuro-fuzzy systems have found extensive applications in pattern recognition [2]. In this context, several techniques for deriving fuzzy rules from training data such as fuzzy clustering and partitioning based methods have been proposed. The fuzzy clustering based methods search the input space for clusters, which are then projected to each dimension of input space to gain fuzzy rules with better interpretability. This approach encompasses a variety of algorithms such as Kohonen learning rule, hyper-box method, product-space partitioning, and fuzzy C-mean method [3]. Examples of partitioning based methods are NEFCLASS and

Armin Eftekhari · Mojtaba Ahmadieh Khanesar · Mohamad Forouzanfar · Mohammad Teshnehlab
Faculty of Electrical Engineering, K.N. Toosi University of Technology, P.O. Box 16315-1355, Tehran, Iran
e-mail: {a.eftekhari,ahmadieh,mohamad398}@ee.kntu.ac.ir, teshnehlab@eetd.kntu.ac.ir

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 305–314.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

NEFCAR, which start with large number of partitions and are then pruned to select the fuzzy rules [2, 4]. For a detailed discussion on neuro-fuzzy rule generation algorithms, the reader is referred to [15- 19].

This study proposes a novel incremental technique for structure optimization of local linear neuro-fuzzy classifiers. The proposed neuro-fuzzy classifier is built starting from the simplest form (a global linear classifier). If the overall performance of the classifier was not satisfactory, it would be iteratively refined by incorporating additional local classifiers. Proposed refinement strategy is motivated by LOLIMOT, which is a greedy partition algorithm for structure training of local linear neuro-fuzzy models that determines the (sub) optimal partitioning of input space by axis-orthogonal splits [5] and has found extensive applications in identification problems due to fast implementation and high accuracy. Adoption of LOLIMOT algorithm to classification requires inevitable modifications. Conventional LOLIMOT is restricted to axis-orthogonal splits and is unable of handling high dimensional data. We address these problems by employing a well-known statistical stage, namely linear discriminant analysis (LDA). Therefore, antecedent structure of neuro-fuzzy classifier is built in the transformed (and if needed reduced) input space by axis-orthogonal splits. Moreover, for proper adoption of LOLIMOT algorithm to classification, a novel interpretation of error is introduced. Once the antecedent parameters are determined, rule consequent parameters are efficiently estimated using a local least square approach. To assess the performance of the proposed method, results are compared with conventional classifiers (neural networks, linear Bayes, and quadratic Bayes), neuro-fuzzy classifiers (NEFCLASS and FuNe I), piecewise linear classifiers and decision trees (C4.5). Experimental results on several well-known datasets demonstrate that, in most cases, our algorithm outperforms state-of-art classifiers and significantly improves the classification results.

2 Local Linear Neuro-Fuzzy Classifier

A neuro-fuzzy system with multiple outputs can be realized either by a single SIMO or MIMO model or by a bank of SISO or MISO models. In the current study, the former approach is pursued as it often requires fewer neurons [5]. Assume a set of input/label pairs $\{U, Y\}$ where $U \in \mathbb{R}^{p\times 1}$ and $Y \in \{0,1\}^{K\times 1}$. In the case of two-class problems, it is most convenient to use the binary representation, in which there is a single target variable $y \in \{0,1\}$ such that $y = 1$ represents first class and $y = 0$ represents the other class. When facing a K-class problem, it is often convenient to use a 1-of-K coding scheme in which the label Y is a vector of length K such that if the class is C_j, then all elements of Y are zero except its jth element denoted by $(Y)_j$, which takes the value 1. The elements of label vector Y can be interpreted as posterior probabilities of corresponding classes, with the values of probability taking only the extreme values of 0 and 1. Therefore, we wish to predict discrete class labels, or more generally posterior probabilities that lie in the range $(0,1)$. This is achieved by introducing an activation function $f(\cdot)$ [20] to limit the output of the model so that it falls into $(0,1)$. The choice of activation

function is usually logistic sigmoid ($K = 2$ classes) or softmax ($K \geq 2$ classes). Decision is made by assigning each test sample to the class with maximum posterior probability. The network architecture of a neuro-fuzzy classifier structured based on local linear classifiers, is depicted in Fig. 1, where the rule antecedent inputs $Z \in \mathbb{R}^{nz\times 1}$ and the rule consequent inputs $X \in \mathbb{R}^{nx\times 1}$ are subsets of the input samples U. Each neuron $i = 1, \dots, M$ of the model realizes a fuzzy rule:

$$R_i = \text{IF } (Z)_1 \text{ IS } A_{i,1} \text{ AND } \dots \text{AND } (Z)_{nz} \text{ IS } A_{i,nz} \text{ THEN } \hat{Y} = W_i^{\mathrm{T}} X \tag{1}$$

where $A_{i,j}$ is the jth fuzzy set defined on ith input and $W_i \in \mathbb{R}^{nx\times K}$. Each neuron or rule represents K local linear classifiers (LLCs) and an associated validity (weighting) function that determines the region of validity of those LLCs. For a reasonable interpretation of local classifiers it is furthermore necessary that the validity functions sum up to one for any antecedent input Z. The output of the local linear neuro-fuzzy classifier would be:

$$\hat{Y}(U) = f\left(\sum_{i=1}^{M} \varphi_i(Z) \cdot \hat{Y}_i(X)\right) = f\left(\sum_{i=1}^{M} \varphi_i(Z) \cdot \left(W_i^{\mathrm{T}} X\right)\right) \tag{2}$$

where $\hat{Y}_i$ denotes the output of local models and $\varphi_i(\cdot)$ is interpreted as weighting function, $i = 1, \dots, M$. Thus, the output of the model is obtained by applying $f(\cdot)$ to the weighted sum of the outputs of the LLCs. In other words, the model interpolates between local models by weighting functions. In the following, the validity functions $\varphi_i(\cdot)$ are chosen to be normalized Gaussians:

$$\varphi_i(Z) = \frac{\mu_i(Z)}{\sum_{j=1}^{M} \mu_j(Z)}, \mu_i(Z) = \exp\left(-\frac{1}{2}(Z - C_i)^{\mathrm{T}} \Sigma_i^{-1} (Z - C_i)\right) \tag{3}$$

where $C_i \in \mathbb{R}^{nz\times 1}$ is the center of ith membership function and $\Sigma_i \in \mathbb{R}^{nz\times nz}$ is a diagonal matrix containing variances of individual dimensions, i.e. $\Sigma_i = \text{diag}\{\sigma_{i,1}^2, \dots, \sigma_{i,nz}^2\}$. Here, we will assume the most general case for X and Z, where $X = Z = U$. It should be pointed out that in contrast to the models used for identification, discussed neuro-fuzzy classifier will be no longer linear in the consequent parameters due to the presence of $f(\cdot)$. This will lead to more analytical and computational complexities than for identification models. However, at the expense of losing the probabilistic point of view, we can omit the nonlinear activation function as in [7]. A test sample is then assigned to the class with maximum activation value and the classifier would be linear in consequent parameters. Optimization of rule antecedent structure and rule consequent parameters are discussed in the following sections.

2.1 Rule Consequent Parameters

Rule consequent parameters are interpreted as parameters of local classifiers. The neuro-fuzzy classifier presented by (2) is linear in consequent parameters, due to linearity assumption for activation function. Therefore, these parameters can be efficiently estimated from training patterns using a least square approach, provided the rule antecedent structure is given. Simultaneous optimization of all consequent parameters (global optimization), yields the best results in the sense of least mean

square error but involves extreme computational effort. Alternatively, we can use local estimation approach presented in [5], which neglects the overlap between the validity functions and estimates the parameters of each rule separately. This approach is computationally more efficient than global estimation. The cost, however, is the introduction of a bias error while on the other hand, the variance error (and the effect of over-fitting) is decreased and more robustness to noise is gained. In this paper, the local estimation approach is pursued and is described as follows. Instead of estimating all $M \times K \times (p+1)$ consequent parameters simultaneously (as in global estimation), M local estimations are carried out for the $K \times (p+1)$ parameters of each neuron. Note that the parameter matrix associated with ith LLC is $W_i \in \mathbb{R}^{(p+1)\times K}$ and that the contribution of ith LLC to the output vector $\hat{Y}_i(U)$ is $\varphi_i(U)\left(W_i^{\mathrm{T}}U\right)$, $i = 1, \ldots, M$. The contribution of ith LLC is dominant only in the region where the associated validity function $\varphi_i(\cdot)$ is close to one (which happens near the center of $\varphi_i(\cdot)$). Training samples in this region are highly relevant for the estimation of W_i. Therefore, local estimation of W_i can be achieved by performing the following weighted least square optimization:

$$\min_{W_i} I_i \quad \text{where } I_i = \sum_{j=1}^{N} \varphi_i\left(U_j\right) \cdot \left\| Y\left(U_j\right) - \hat{Y}\left(U_j\right) \right\|^2 \tag{4}$$

where U_j denotes the jth input sample, $j = 1, \ldots, N$. This optimization is equivalent to fitting a linear classifier to weighted training data. Let the target matrix $\Upsilon \in \mathbb{R}^{N\times K}$, the regression matrix $X \in \mathbb{R}^{N\times(p+1)}$, and the weighting matrix $Q_i \in \mathbb{R}^{N\times N}$ be defined as follows:

$$\Upsilon = [Y_1 \;\; Y_2 \;\; \cdots \;\; Y_N]^{\mathrm{T}}, \; X = \begin{bmatrix} 1 & U_1^{\mathrm{T}} \\ 1 & U_2^{\mathrm{T}} \\ \vdots & \vdots \\ 1 & U_N^{\mathrm{T}} \end{bmatrix}, \; Q_i = \begin{bmatrix} \varphi_i(U_1) & 0 & \cdots & 0 \\ 0 & \varphi_i(U_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \varphi_i(U_N) \end{bmatrix} \tag{5}$$

Then, it can be simply verified that the optimum $W_i \in \mathbb{R}^{(p+1)\times K}$ that minimizes (4) is obtained as:

$$W_i = (X^{\mathrm{T}} Q_i X)^{-1} X^{\mathrm{T}} Q_i \Upsilon \tag{6}$$

2.2 Rule Antecedent Structure

Training of the antecedent parameters is a nonlinear optimization task, which provides us with the proper partitioning of the input space. Two common strategies for antecedent structure optimization are clustering and partitioning based techniques. In order to embed data-driven knowledge in a neuro-fuzzy system, clustering methods such as Fuzzy RuleNet [8] utilize cluster vectors extracted from the input dataset to initialize the centers of fuzzy rules. A learning algorithm is then applied to fine tune the rules based on the available training data. These approaches usually search for hyper-ellipsoidal or hyper-rectangular clusters in input space and are shown to produce rules which are hard to interpret. Partitioning based methods such as NEFCLASS [4] divide the input space into finer regions by grid partitioning. Each partition is supposed to represent an If-Then rule. These rules are then pruned using some heuristics. Finally, membership functions are defined

using only best performing rules. In other words, NEFCLASS does not induce fuzzy classification rules by searching for clusters, but by modifying the fuzzy partitionings defined on each single dimension. Evidently, partition based approaches are computationally expensive [9].

3 Proposed Algorithm for Structure Optimization

As discussed in subsection 2.1, tuning of the antecedent parameters of a neuro-fuzzy system is a nonlinear optimization task. Optimization of consequent parameters by local least square approach was also discussed in Subsection 2.2. The proposed algorithm for structure optimization increases the complexity of the local linear neuro-fuzzy classifier during the training phase. Hence, it starts with a coarse partitioning of the input space, which is then refined by increasing the resolution of the input space partitioning. The proposed algorithm is based on divide-and-conquer principle. Our strategy for input space partitioning is motivated by LOLIMOT, which is a local linear neuro-fuzzy algorithm that uses axis-orthogonal splits to avidly partition the input space. Computational complexity of LOLIMOT grows linearly with the number of neurons and cubically with the number of consequent parameters of each neuron. This level of computation complexity is quite favorable [5]. As will be discussed shortly, adoption of LOLIMOT algorithm to classification requires inevitable modifications.

One of the most severe restrictions of LOLIMOT is the axis-orthogonal partitioning of the input space. This restriction, while being crucial for interpretation as a fuzzy system and for the development of an extremely efficient construction algorithm, leads to the following shortcomings: (i) improper splitting of input space, which frequently happens when optimal partitioning of input space does not align with axis-orthogonal directions. In such cases, the nonlinearity of data in the original input space does not stretch along the input space axes and hence LOLIMOT cannot efficiently determine proper input partitioning [10]. (ii) curse of dimensionality, which often plagues fuzzy systems in real world applications. At each iteration, LOLIMOT tries all divisions of worst LLC to decide about further refinement so curse of dimensionality will be more prohibitive. Several techniques have been proposed to address these two drawbacks. For example, Nelles developed an axis-oblique decomposition algorithm, which suffers from computational concerns [10]. In addition, using different input spaces for rule antecedent and consequent was suggested in [5], which could result in the alleviation of computational efforts. Evidently, adoption of LOLIMOT to classification confronts the above shortcomings, especially when the discriminancy of classes is small in the original axes. We suggest using a computationally cheap, easy to implement statistical stage, namely LDA [3]. The basic concept of LDA is to seek for the most efficient projective directions which minimize the scattering of samples of the same class and maximize the distance of different classes. In addition, LDA is capable of selecting the best linear combinations of input features for classification and hence can be used for dimensionality reduction. Therefore, axis-orthogonal partitioning of the transformed input space (building the structure in

the transformed space), often results in significant reduction in the complexity of antecedent structure, as well as computational cost.

Another practical concern for adoption of LOLIMOT to classification is the error interpretation. Training of local linear models in a LOLIMOT system is achieved by minimizing the local loss function defined in (4). This loss function is also used for comparison of local models. In this paper, a novel interpretation of error is introduced. While the loss function of (4) is used to train the LLCs, we suggest using a different error index for comparison of LLCs, which is based on percentage error rather than l_2-norm ($\|\cdot\|_2$) of the classification error. The percentage error resembles the l_1-norm of error ($\|\cdot\|_1$), which is shown to gain better classification results than l_2-norm of error [11]. Through our experiments, it was found that this interpretation of error improves the classification results.

Finally, note that the standard deviation of validity functions is $1/3$ of the length of hyper-rectangle in each dimension [5]. The proposed algorithm can be summarized as follows:

1. *Finding the most discriminative basis*: apply LDA in order to find the most discriminative basis. If needed, dimension reduction is also realized in this step by keeping only the most discriminative features in the new basis. The antecedent structure is built in this transformed space.
2. *Start with an initial model*: use any prior knowledge to construct the validity functions in the transformed initial input space partitioning. If no input space partitioning is available a priori, then set $M = 1$ and start with a single LLC.
3. *Compare LLCs to find the worst LLC*: use the following equation to calculate the error index l for all LLCs, in which each misclassified pattern is assigned to the LLC with largest degree of validity. Then, the LLC with maximum error index l is selected as the worst performing, which is denoted by $\mathrm{LLC_b}$.

$$l_i = \frac{N(S_{i,e})}{N(S_i)} \text{ where } \quad S_i = \{U | \varphi_i(U) > \varphi_j(U) \text{ for } j \neq i\}$$

$$S_{i,e} = \{U | U \in S_i \text{ AND } U \text{ is misclassified}\} \tag{7}$$

   where, $N(\xi)$ denotes the number of elements of vector ξ.
4. *Check all divisions of worst LLC*: consider the $\mathrm{LLC_b}$ for further refinement. The hyper-rectangle of this LLC is partitioned into two halves with an axis-orthogonal split. Divisions in all dimensions are considered. For each of the p divisions, the following steps are taken.
   a. Construct the membership functions for both hyper-rectangles.
   b. Construct all validity functions.
   c. For both newly generated LLCs, weigh the training samples with corresponding validity functions and fit a linear classifier to these weighted samples by minimizing local loss function defined in (6) (local optimization of the rule consequent parameters for both newly generated LLCs).
   d. Calculation of the overall error index $l = \sum_{i=1}^{M} l_i$ for the current model using (7).

5. *Find the best division*: select the best of the p alternatives checked in Step 4. The validity functions constructed in Step 4a and the LLCs optimized in Step 4c are included in the classifier. The number of LLCs is increased by one.
6. *Test for convergence*: if the termination criterion (e.g. convergence of performance) is not met, go to Step 2.

Fig. 2 depicts the partitioning of input space obtained by applying our algorithm to Iris dataset, which is a multivariate dataset consisting of 50 samples from three species of Iris flowers. By applying principal component analysis, number of features was reduced to two (for better visualization) and one half of samples were randomly selected for training. Note that the splitting directions are not axis-orthogonal, but are selected to maximize the linear discriminancy of training samples. Each rectangle shows the validity region of corresponding LLC.

4 Experiments

This section presents the classification results of the proposed method on several well-known datasets. The error rates of the proposed classifier are compared to that of a number of existing pattern classification algorithms. To this end, four datasets from ELENA [21] project, namely Iris_CR, Phoneme_CR, Satimage_CR, and Texture_CR and two datasets from UCI machine learning repository [22], namely Wisconsin breast cancer and Sonar are selected. ELENA project and UCI machine learning repository are resources of databases for testing and benchmarking pattern classification algorithms. The CR affix in the names of datasets from ELENA project indicates that the dataset is initially preprocessed by a normalization routine to center each feature and enforce unit variance. In our experiments on these datasets, we follow a similar technique to [12]. First, each dataset is partitioned into two equal random sets: one for training and the other for test phase. Then, the roles of two halves are reversed. To achieve more accurate results, experiments are performed 20 times and the average error rate is reported.

4.1 Comparison with Conventional Classifiers

Table 1 lists average classification error rates of the proposed algorithm and of some conventional classifiers, namely neural network, linear Bayes, and quadratic Bayes on several datasets, as reported in [12]. In [12], each classifier has been reasonably optimized with regards to parameter settings and available features. In addition, an earnest effort was made to optimize each individual classifier with respect to selecting good values for the parameters which govern its performance. Moreover, feature selection techniques have been applied to feed each classifier with best features. Results indicate that the proposed simple local linear fuzzy classifier can be quite successful compared to these conventional classifiers. Note that the proposed algorithm achieves better results than the neural network classifier which can be assumed as a close relative of the proposed algorithm.

Table 1 Error rates in percentage for conventional and proposed classifiers on several datasets. The best results are highlighted in boldface

	Datasets			
	Iris_CR	Satimage_CR	Texture_CR	Phoneme_CR
Neural network	4.67	16.02	5.15	**20.79**
Linear Bayes	2.67	16.69	2.58	27.00
Quadratic Bayes	4.67	14.22	**0.96**	24.59
Proposed classifier	**2.33**	**13.54**	2.80	23.15

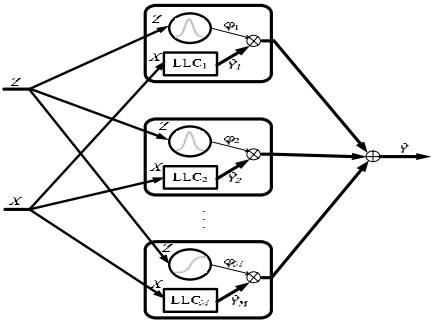


Fig. 1 Local linear neuro-fuzzy classifier

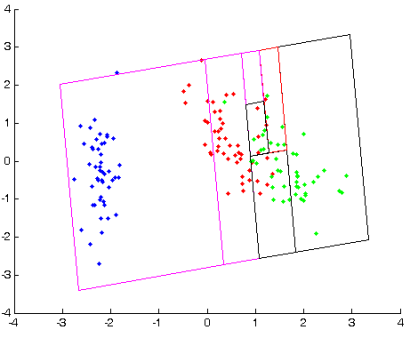


Fig. 2 Operation of proposed algorithm on Iris dataset

4.2 Comparison with Piecewise Linear Classifiers

Piecewise linear classifiers aim at approximating complex decision boundaries with piecewise linear functions. Recently, Kostin presented a simple and fast piecewise linear classifier which demonstrated comparable (even superior in many cases) results with many well-known benchmark classifiers [13]. The motivation for this experiment is the similarities of our classifier and this family of classifiers. The average classification error rates for two methods are listed in Table 2.

Table 2 Error rates for piecewise linear classifier [6], C4.5 decision tree [14] and proposed classifier on several datasets. The best results are highlighted in boldface

	Datasets					
	Iris_CR	Satimage_CR	Texture_CR	Phoneme_CR	BCW	Sonar
Piecewise linear	3.34	13.90	4.90	17.85	4.80	19.70
C4.5	7.33	16.50	11.91	**16.08**	5.26	25.60
Proposed classifier	**2.33**	**13.54**	**2.80**	23.15	**2.84**	**10.71**

4.3 Comparison with Decision Tree Classifiers

Due to the same essence and characteristics, comparison of our proposed classifier and decision tree classifiers is inevitable. Here, we compare our method with C4.5 which is a well-known decision tree classifier [14]. Table 2 lists average classification error rates of our proposed method as well as C4.5 classifier.

5 Conclusions

In this study, a simple and computationally efficient local linear neuro-fuzzy classifier has been introduced, implemented and tested on a number of well-known datasets. The structure of the antecedent part is obtained during the training phase and is data driven rather than knowledge based. Input space is first transformed by LDA, so that the linear discriminancy of training samples is maximized. The antecedent structure is then built in the transformed space by axis-orthogonal splits. At each iteration, the local linear fuzzy classifier with the worst error index is split into two new rules which are then included in the classifier. In addition, the rule consequent parameters are optimized using a local least square approach. The simplicity and speed are the main advantages of the proposed classifier. Together with high performance, this classifier is a good choice for many applications in which the use of more sophisticated classifiers can be impractical.

References

[1] Czogala, E., Leski, J.: Fuzzy and Neuro-Fuzzy Intelligent Systems. Springer Soft computing (2000) ISBN: 3790812897

[2] Taur, J.S., Tao, C.W.: A New Neuro-Fuzzy Classifier with Application to On-Line Face Detection and Recognition. Journal of VLSI Signal Processing 26, 397–409 (2004)

[3] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, Chichester (2001)

[4] Nauck, D., Kruse, R.: NEFCLASS A Neuro-Fuzzy Approach for the Classification of Data. In: ACM Symposium on Applied Computing (1995)

[5] Nelles, O.: Nonlinear System Identification from Classical Approaches to Neural Networks and Fuzzy Models. Springer, Heidelberg (2002)

[6] Kostin, A.: A Simple and Fast Multi-Class Piecewise Linear Pattern Classifier. Pattern Recognition 39, 1949–1962 (2006)

[7] Keles, A., Hasiloglu, S., Keles, A.: Neuro-fuzzy Classification of Prostate Cancer using NEFCLASS-J. Computers in Biology and Medicine 37, 1617–1628 (2007)

[8] Tschichold-Gfirman, N.: Generation and Improvement of Fuzzy Classifiers with Incremental Learning Using Fuzzy Rulenet. In: Proceeding of the ACM Symposium on Applied Computing, pp. 466–470 (1995)

[9] Nauck, D., Kruse, R.: A neuro-fuzzy method to learn fuzzy classification rules from data. Fuzzy Sets and Systems 89, 277–288 (1997)

[10] Nelles, O.: Axes-oblique Partitioning Strategies for Local Model Networks. In: IEEE International Symposium on Intelligent Control, vol. 9, pp. 2378–2383 (2006)

[11] Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust Face Recognition via Sparse Representation. To appear on IEEE Transactions on Pattern Analysis and Machine Intelligence
[12] Kegelmeyer, W.P., Woods, K., Bowyer, K.: Combination of Multiple Classifiers Using Local Accuracy Estimates. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(4) (1997)
[13] Halgamuge, S.K., Glesner, M.: Neural Networks in Designing Fuzzy Systems for Real World Applications. Fuzzy Sets and Systems 65, 1–12 (1994)
[14] Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann publishers Inc., San Francisco (1993)
[15] Lin, C.T., Yeh, C.M., Liang, S.F., Chung, J.F., Kumar, N.: Support-Vector-Based Fuzzy Neural Network for Pattern Classification. IEEE Transactions on Fuzzy Systems 14, 31–41 (2006)
[16] Wu, S., Er, M.J., Gao, Y.: A Fast Approach for Automatic Generation of Fuzzy Rules by Generalized Dynamic Fuzzy Neural Networks. IEEE Transactions of Fuzzy Systems 9(4), 578–594 (2001)
[17] Ishibuchi, H., Nakashima, T.: Effect of Rule Weights in Fuzzy Rule-Based Classification Systems. IEEE Transactions on Fuzzy Systems 9(4), 506–515 (2001)
[18] Nozaki, K., Ishibuchi, H., Tanaka, H.: Adaptive Fuzzy Rule-Based Classification Systems. IEEE Transactions on Fuzzy Systems 4, 238–250 (1996)
[19] Mitra, S., Hayashi, Y.: Neuro-fuzzy Rule Generation: Survey in Soft Computing Framework. IEEE Transactions on Neural Networks 11, 748–768 (2000)
[20] McCullagh, P., Nelder, J.A.: Generalized Linear Models, 2nd edn. Chapman and Hall, Boca Raton (1989)
[21] Databases of ELENA project, `http://ftp.dice.ucl.ac.be/pub/neural-nets/ELENA/databases`
[22] Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases, University of California, Department of Information and Computer Science, Irvine, CA (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`

On Criticality of Paths in Networks with Imprecise Durations and Generalized Precedence Relations

Siamak Haji Yakhchali, Seyed Hassan Ghodsypour,
and Seyed Mohamad Taghi Fatemi Ghomi

Abstract. This research deals with problems of the criticality of paths in networks with generalized precedence relations (GPRs) and imprecise activity and time lag durations, represented by means of interval or fuzzy numbers. So far, these problems have been considered when networks have classical finish-start precedence relations by several authors. However, in practice it is often necessary to specify other than the finish-start precedence relations. Proposed theorems ascertain whether a given path is necessarily critical, possibly critical or necessarily non-critical in interval-valued networks with GPRs. The results are extended to networks with fuzzy activity and time lag durations and novel linear programming models are proposed to calculate the degree of necessity and possibility that a given path is critical.

1 Introduction

The Critical Path Method (CPM) [10] is devoted to minimizing the makespan of a project. Finish-start relations with zero time lag durations between activities was assumed in CPM. Over the years, this assumption has been relaxed. The resulting types of precedence relations will be referred to generalized precedence relations (GPRs) [8]. GPRs consist of four different types: start-start (SS), start-finish (SF), finish-start (FS) and finish-finish (FF). GPRs can specify a minimal or a maximal time lag between a pair of activities. A minimal time lag states that an activity can only start (finish) when the predecessor activity has already started (finished) for a certain time period. A maximal time lag specifies that an activity should be started (finished) at the latest, within a certain number of time periods beyond the start (finish) of another activity. GPRs can be used to model a wide variety of specific problem characteristics, including: activity ready times and deadlines, activities that have to start or terminate simultaneously, non-delay execution of activities, mandatory activities overlaps, fixed activity start times, set-up times and etc [5].

However in the real world, the activity and time lag duration are usually difficult to estimate precisely. In these situations, the fuzzy set scheduling literature recommends the use of fuzzy numbers for modeling activity duration. Fuzzy critical path methods have been proposed since the late 1970s (e.g. [12]). Forward

Siamak Haji Yakhchali · Seyed Hassan Ghodsypour · Seyed Mohamad Taghi Fatemi Ghomi
Department of Industrial Engineering, Amirkabir University of Technology,Tehran, Iran
e-mail: {yakhchali,ghodsypo,fatemi}@aut.ac.ir

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 315–324.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

recursion, comparable to the one used in classical CPM, could determine the possible values of the earliest starting times but backward recursion fails to compute the possible values of the latest starting times. Several authors tried to cope with this problem. Zielinski [20] has determined the possible values of the latest starting times of activities by proposing polynomial algorithms. Dubois et al. [6] have proposed an algorithm based on path enumeration to compute optimal intervals for latest starting times and floats. Fortin et al. [9] have provided a solution to the problem of finding the maximal floats of activities and Yakhchali and Ghodsypour [15] have proposed a hybrid genetic algorithm for the problem of finding the minimal float of an activity.

The criticality concept in networks with interval (fuzzy) activities durations is a more realistic approach than the traditional ones. Instead of being critical or not, the activities or paths that are for sure critical despite uncertainty, is called necessarily critical, those that are for sure not critical is called necessarily noncritical and those whose criticality is unknown, called possibly critical [2]. The problems of the necessarily and possibly critical paths in the networks with imprecise activity and time lag durations have been discussed by Yakhchali et al. [13], [14].

More recent research has been directed at relaxing both of deterministic activity and time lag durations and strict finish-start precedence relations. Yakhchali and Ghodsypour have solved the problem of determining the possible value of the latest starting time of an activity in acyclic networks [16] and cyclic networks [17] with GPRs and imprecise durations. They suggested an algorithm for computing both the possible values of the latest starting times and the floats of all activities in these networks [18]. Although temporal analysis in networks with imprecise durations and generalized precedence relations was discussed, the criticality of paths in such networks was not addressed in the literature thus far. So, the criticality analysis carried out on the basis of general precedence relations and imprecise durations was incomplete. In the following, the criticality of paths in the network with interval and fuzzy durations under generalized precedence relation is provided.

2 Terminology and Representation

The project scheduling problems to be dealt with throughout this paper can be stated as follows. A set *V={1,2, …, n}* of *activities* has to be executed whereactivity durations $i \in V$ are chosen from intervals $D_i = [\underline{d}_i, \overline{d}_i]$, $\underline{d}_i \geq 0$. Thenon-preemptable activities are numbered from *1* to *n*, where the dummy activities *1* and *n* represent the beginning and the termination of the project, respectively. Activities can be represented by an *activity-on-node (AON)* network *G=<V, E >* with node set *V*, arc set *E*. The arc set or generalized precedence relations, *E*,consist of *minimal or maximal time lag*. If between two activities *i* and *j* aminimal or maximal time lag is prescribed, we introduce an arc (*i, j*) fromnode *i* to node *j* weighted by an interval number which have theforms: $s_i + SS_{ij}^{min} \leq s_j \leq s_i + SS_{ij}^{max}$, $s_i + SF_{ij}^{min} \leq f_j \leq s_i + SF_{ij}^{max}$, $f_i + FS_{ij}^{min} \leq s_j \leq f_i + FS_{ij}^{max}$ and $f_i + FF_{ij}^{min} \leq f_j \leq f_i + FF_{ij}^{max}$ where the start of an

activity is given by s_i and its finishing time denoted by f_i. SS_{ij}^{min} represents a minimal time lag between the start time of activity *i* and the start time of activity *j* and the value of SS_{ij}^{min} are chosen from intervals $SS_{ij}^{min} = [\underline{ss}_{ij}^{min}, \overline{ss}_{ij}^{min}]$. Analogously SS_{ij}^{max},..., FF_{ij}^{max} as well as SS_{ij}^{min} are defined.

The various time lags can be represented in a *standardized form* by transforming them to, for instance, minimal SS precedence relation, using the transformation rules. The transformation rules for the network *G* with interval activity and time lag durations were proposed by Yakhchali and Ghodsypour [17], [18]. In this way, all GPRs are consolidated in the expression $s_i + L_{ij} \le s_j$. The minimum time lag, denoted by $L_{ij} = [\underline{l}_{ij}, \bar{l}_{ij}]$, *i, j*$\in V$, implies that *j* can start l_{ij} which is chosen from intervals $[\underline{l}_{ij}, \bar{l}_{ij}]$ units of time after the start of *i* at the earliest.

The notation of *configuration* denoted by Ω has been defined to relate the interval case to the deterministic case of classical CPM problems. A configuration is a tuple of time lag durations, (l_{12},...,l_{ij}), such that $\forall (i,j) \in E$, $l_{ij} \in L_{ij}$. For a configuration Ω, $l_{ij}(\Omega)$ will denote the duration of time lag (*i, j*). Let ω be the set of possible configurations of time lag durations.

It should be noted that the network, *G*, may contain cycles. A path $\langle i_s, i_k, i_l, ..., i_t \rangle$ is called a *cycle* if $s = t$. "*Path"* and "*cycle*" mean a directed path and cycle, respectively. A path is called a *real path* if it does not contain any cycle and similarly a path is called a *fictitious path* if it contains a cycle or cycles. It is obvious that an activity appear only one time in a real path.

The length of a path in a standardized network, denoted by $W_p(\Omega)$, is defined as the sum of all the time lags associated with the arcs belonging to that path. A positive path length from a node to itself indicates the existence of a cycle of positive length, and consequently, the non-existence of a time feasible schedule. So, the network, *G*, is feasible if there exists a feasible schedule of the activities. Otherwise, the activity network is infeasible.

Let us denote the set of all paths in *G* from node "*1*"to "*n*" by *P*. For a path, $p \in P$, we define two configurations. The pessimistic configuration induced by *p*, denoted by Ω_p^+, is a configuration $\Omega_p^+ \in \omega$ such that:

$$l_{ij}(\Omega_p^+) = \begin{cases} \bar{l}_{ij} & \text{for } (i,j) \in p \\ \underline{l}_{ij} & \text{for } (i,j) \notin p \end{cases} \tag{1}$$

Similarly Ω_p^-, called the optimistic configuration induced by *p*, is defined as formula (2).

$$l_{ij}(\Omega_p^-) = \begin{cases} \underline{l}_{ij} & \text{for } (i,j) \in p \\ \bar{l}_{ij} & \text{for } (i,j) \notin p \end{cases} \tag{2}$$

3 Necessary Criticality in Interval Networks with GPRs

An activity is necessarily critical in G if and only if its maximal float is null [6]. The problem of evaluating the necessary criticality of a given activity $k \in V$ in the interval network with GPRs completely has been solved [19].

Let us consider the problem of ascertaining the necessary criticality of a given path in the network G, which did not discuss in the literature. Theorem 1 provides necessary and sufficient conditions for determining the necessary criticality of $p \in P$.

Theorem 1. A path $p \in P$ is necessarily critical in the network G if and only if it is critical in the usual sense in Ω_p^-, the optimistic configuration induced by p.

Proof: The *only if* direction ($\Rightarrow$): Let $p \in P$ be necessarily critical path in G. From necessary criticality of p, it follows that for each configuration, $\forall \Omega \in \omega$, p is critical in the usual sense.

The *if* direction ($\Leftarrow$): We must show that for each configurations, the path p is the longest path in the network G. Assume on the contrary that there exists a configuration, $\Omega' \in \omega$, such that $W_p(\Omega') < W_{p'}(\Omega')$ where $p' \in P$. It means that p is not critical in Ω' so p is not necessary critical. According to the definition of Ω_p^-, we know that the length of p in Ω_p^- is less than or equal to its length in other configurations, thus $W_p(\Omega_p^-) \le W_p(\Omega')$. Consequently, we have $W_p(\Omega_p^-) < W_{p'}(\Omega_p^-)$. This fact contradicts that p is critical in the usual sense in Ω_p^-. Hence, p is necessary critical. □

Theorem 1 reduces the number of possible configurations to the configuration Ω_p^-. Thus the problem of ascertaining if a given path $p \in P$ is necessary criticality reduces to applying the classical GPRs methods, like Floyd-Warshall algorithm [11] (time complexity $O(|V|^3)$) and Modified Label Correcting algorithm [1] (time complexity $O(|V||E|)$) for the configuration Ω_p^-.

Note that in the case of p is not critical in the usual sense in Ω_p^-, it is impossible to judge its type of criticality, i.e. p may be the possibly critical path or the necessarily non-critical path. This topic discusses in the next section.

4 Possible Criticality in Interval Networks with GPRs

This section deals with the problems related to possible criticality in the network G. An activity $k \in V$ is possibly critical in G if and only if its minimal float is null

[6]. Unfortunately the problem of ascertaining the possibly critical activity in a network with interval activity durations and classical precedence relations is strongly *NP*-complete [4]. Thus it may be concluded that the problem of ascertaining the possibly critical activity in the network with interval activity and time lag durations and GPRs is at least as hard, if not harder than, the same problem in the network with classical precedence relations. Accordingly, we focus on the problem of ascertaining the possibly critical path.

4.1 Necessarily Non-critical Path

Let us consider the problem of ascertaining the necessary non-criticality of a given path in the network *G*, before possibly critical path. Theorem 2 gives necessary and sufficient conditions for ascertaining the necessary non-criticality of $p \in P$ in the network *G*.

Theorem 2. A path $p \in P$ is necessarily non-critical in the network *G* if and only if it is not critical in the usual sense in Ω_p^+, the pessimistic configuration induced by *p*.

Proof: The *only if* direction ($\Rightarrow$): Let $p \in P$ be necessarily non-critical path in *G*. From necessary non-criticality of *p*, it follows that for each configuration, $\forall\, \Omega \in \omega$, *p* is not critical in the usual sense. Thus *p* is not critical in the usual sense in Ω_p^+.

The *if* direction ($\Leftarrow$): We conduct an indirect proof. Let us assume that there exists a configuration $\Omega' \in \omega$ such that the given path *p* is critical in this configuration, Ω'.

Thus, the of *p* in Ω' is greater than or equal to the length of each path, i.e. $W_p(\Omega') \geq W_{p'}(\Omega')$, $p' \in P$. If we increase the time lag durations $l_{ij}(\Omega')$ to $\bar{l}_{ij}$ of all $(i,j) \in p$ and decrease the time lag durations $l_{ij}(\Omega')$ to $\underline{l}_{ij}$ of all $(i,j) \notin p$, the path *p* will remain the longest path in *G* for this new configuration or the pessimistic configuration induced by *p*, Ω_p^+. Consequently, we have $W_p(\Omega_p^+) \geq W_{p'}(\Omega_p^+)$. This fact contradicts that *p* is not critical in the usual sense in Ω_p^+. Hence, *p* is necessary non-critical. □

According to Theorem 2, the problem of ascertaining the given path is necessarily non-critical reduces to applying the Floyd-Warshall or Modified Label Correcting algorithm, for the configuration Ω_p^+.

4.2 Possibly Critical Path

Theorem 3 provides necessary and sufficient conditions for establishing the possible criticality of a given path $p \in P$ in the network *G*.

Theorem 3. A path $p \in P$ is possibly critical in the network G if and only if it is not critical in Ω_p^- and it is critical in Ω_p^+, in the usual sense.

Proof: The *only if* direction ($\Rightarrow$): Straightforward. Based on the definition of a possibly critical path, there exist a configuration, $\Omega' \in \omega$, which p is critical in it and also a configuration, $\Omega'' \in \omega$, that p is non-critical in it. The length of p in Ω_p^+ is greater than or equal to its length in Ω' and the length of other paths in Ω_p^+ is less than or equal to their length in Ω', so p is critical in Ω_p^+. In the similar manner, p is not critical in Ω_p^-.

The *if* direction ($\Leftarrow$): This fact deduces directly from the definition of a possibly critical path. Since there exist the pessimistic configuration induced by p, Ω_p^+, in which p is critical and the optimistic configuration induced by p, Ω_p^-, in which p is non critical. □

From Theorem 3 it follows that the running time of the problem of ascertaining the given path is possibly critical is $O(|V||E|)$, the running time of Modified Label Correcting algorithm.

5 Criticality in Networks with Fuzzy Durations and GPRs

All the elements of the network G are the same as in the interval case except for the activity and time lag durations which are determined by means of fuzzy numbers. Fuzzy numbers express uncertainty connected with the ill-known activity and time lag durations modeled by these numbers which generate possibility distributions for the sets of values containing the unknown activity and time lag durations. $\tilde{D}_i$ will represent the fuzzy number related to the possible duration of i, $i \in V$ and $S\tilde{S}_{ij}^{min}$ denotes the fuzzy number that contains possible value of the minimal time lag between the start time of activity i and the start time of activity j (similar definition apply for $S\tilde{S}_{ij}^{max}$,…). Intuitively, when the durations are fuzzy numbers, the starting time of an activity i, $\tilde{s}_i$, and its finishing time, $\tilde{f}_i$, become fuzzy numbers. The standardized form of the network with fuzzy GPRs can be obtained by proposed transformation rules [17]. In this way, all Fuzzy GPRs are consolidated in the expression $\tilde{s}_i \oplus \tilde{L}_{ij} \le \tilde{s}_j$, where $\tilde{L}_{ij}$ is the fuzzy number.

Let Ω be a configuration of time lag durations in the standardized network. Thus, the (joint) possibility distribution over configurations, denoted by $\pi(\Omega)$, is determined by $\pi(\Omega) = min_{(i,j)\in E}\, \mu_{\tilde{L}_{ij}}(l_{ij})$. Hence, the degree of necessity and possibility that a path, $p \in P$, is critical [3], [2].

$$Poss(\ p \text{ is critical }) = \sup_{\Omega: p \text{ is critical } \Omega} \pi(\Omega) = sup\{\ \lambda \mid p \text{ is possibily critical in } G^{\lambda}\ \} \tag{3}$$

$$Nec(\ p \text{ is critical }) = 1 - \sup_{\Omega: p \text{ is not critical in } \Omega} \pi(\Omega) = sup\{\ \lambda \mid p \text{ is necessarily critical in } G^{1-\lambda}\ \} \tag{4}$$

$$Nec(\ p \text{ is noncritical }) = 1 - Poss(\ p \text{ is critical }) \tag{5}$$

where G^{λ} denotes the $\lambda - cut$ of the network G, i.e., the network G with the interval durations $\tilde{l}_{ij}^{\lambda} = [\underline{l}_{ij}^{\lambda}, \overline{l}_{ij}^{\lambda}]$, $(i, j) \in E$.

Based on the above formulas, the main difficulty is interval valued case instead of lying in the introduction of fuzzy sets which are represented ill-know durations. Thus, the idea of bisection of the unit interval of possible values of λ can be applied to determine the above quantities (see [2] and [3]).

Another effective method for determining the degree of possibility and necessity that a path is critical is based on linear programming which is valid under certain assumption about membership functions of fuzzy durations. Before we pass on to the basic consideration let us recall the notion of a fuzzy number of the *L-R*. A fuzzy number $\tilde{A}$ is called a fuzzy number of the *L-R type* if its membership function $\mu_{\tilde{A}}$ has the following form:

$$\mu_{\tilde{A}}(x) = \begin{cases} 1 & \text{for } x = [\underline{a}, \overline{a}], \\ L(\dfrac{\underline{a} - x}{\alpha_A}), & \text{for } x < \underline{a}, \\ R(\dfrac{x - \overline{a}}{\beta_A}), & \text{for } x > \overline{a}, \end{cases} \tag{6}$$

where *L* and *R* are continuous nonincreasing functions, defined on $[0,+\infty)$, strictly decreasing to zero in those subintervals of interval $[0,+\infty)$ in which they are positive, and fulfilling the conditions *L(0)=R(0)=1*. The parameters α_A and β_A are nonnegative real numbers. The fuzzy number of *L-R type* is denoted by $A = (\underline{a}, \overline{a}, \alpha_A, \beta_A)_{L-R}$ [7]. For a fuzzy number $\tilde{A}$ of *L-R type* $\lambda - cut\ \tilde{A}^{\lambda}$, $\lambda \in (0,1]$, has the form of $\tilde{A}^{\lambda} = [\underline{a} - L^{-1}(\lambda)\alpha_A, \overline{a} + R^{-1}(\lambda)\beta_A]$ where L^{-1} (similarly R^{-1}) denotes the reverse function to *L* in this part of its domain in which it is positive.

Assume that the membership function of the time lag durations in standardized form *G* is given by means of fuzzy number of the *L-R* type, in which the left shape function is equal to the right shape function and additionally the left shape function, *L*, is the same for all durations. In this case $\lambda - cuts$ of a fuzzy time lag $\tilde{L}_{ij}$, $\lambda \in (0,1]$, have the form $\tilde{L}_{ij}^{\lambda} = [\underline{l}_{ij}^{\lambda}, \overline{l}_{ij}^{\lambda}] = [\underline{l}_{ij} - L^{-1}(\lambda)\ \alpha_i, \overline{l}_{ij} + L^{-1}(\lambda)\ \beta_i]$.

Assume that $\theta = L^{-1}(\lambda)$ to developed the linear model by giving the advantage of the fact that function $L^{-1}(y)$ is decreasing in the [0, 1] interval. Thus, time lag

interval durations in the network G^{λ} for a fixed $\lambda \in (0,1]$ is equal $\tilde{L}_{ij}^{\lambda} = [l_{ij}^{\lambda}, \bar{l}_{ij}^{\lambda}] = [\underline{l}_{ij} - \alpha_{ij}\theta, \bar{l}_{ij} + \beta_{ij}\theta]$.

a) Calculating the Degree of Necessity That a Path Is Critical: Based on Theorem 1, a path $p \in P$ is necessarily critical in $G^{1-\lambda}$ can be express as the following equalities and inequalities:

$$\begin{aligned}
&s_i + \bar{l}_{ij} + \beta_{ij}\theta \le s_j \quad \forall (i,j) \in E \text{ such that } (i,j) \notin p \\
&s_i + \underline{l}_{ij} - \alpha_{ij}\theta = s_j \quad \forall (i,j) \in E \text{ such that } (i,j) \in p \\
&s_1 = 0, \\
&s_i \ge 0, \quad \forall i \in V
\end{aligned} \tag{7}$$

If the above model has a solution, p is necessarily critical in $G^{1-\lambda}$. Hence, the following linear programming model calculates the necessity degree of criticality of p, Nec (p is critical).

$$\begin{aligned}
&Maximize \{\theta - \sum_{i \in V} s_i\}, \\
&s_i + \bar{l}_{ij} + \beta_{ij}\theta \le s_j \quad \forall (i,j) \in E \text{ such that } (i,j) \notin p \\
&s_i + \underline{l}_{ij} - \alpha_{ij}\theta = s_j \quad \forall (i,j) \in E \text{ such that } (i,j) \in p \\
&s_1 = 0, \\
&s_i \ge 0, \quad \forall i \in V \\
&\underline{\theta} \le \theta < \bar{\theta}.
\end{aligned} \tag{8}$$

where $\underline{\theta} = L^{-1}(1)$, $\bar{\theta} = L^{-1}(0)$. Thus, Nec (p is critical) $= 1 - L(\theta)$.

As far as the linear programming model is concerned, the term $minimize\{\sum s_i\}$ in objective function is need because without this term linear programming model give earliest starting times for activities that belong to the given path other activities obtain a starting time which is not same as earliest starting time.

b) Calculating the Degree of Possibility That a Path Is Critical: According to Theorem 2-3, a path $p \in P$ is possibly critical in G^{λ} can be express as the following equalities and inequalities:

$$\begin{aligned}
&s_i + \underline{l}_{ij} - \alpha_{ij}\theta \le s_j \quad \forall (i,j) \in E \text{ such that } (i,j) \notin p \\
&s_i + \bar{l}_{ij} + \beta_{ij}\theta = s_j \quad \forall (i,j) \in E \text{ such that } (i,j) \in p \\
&s_1 = 0, \\
&s_i \ge 0, \quad \forall i \in V
\end{aligned} \tag{9}$$

If the above model has a solution, p is possibly critical in G^{λ}. Hence, the following linear programming model calculates the degree of possibility that a path is critical, $Poss$(p is critical).

$$
\begin{aligned}
&\textit{Minimize}\ \{\theta + \sum_{i \in V} s_i\}\,, \\
&s_i + \underline{l}_{ij} - \alpha_{ij}\theta \le s_j \quad \forall (i,j) \in E \text{ such that } (i,j) \notin p \\
&s_i + \bar{l}_{ij} + \beta_{ij}\theta = s_j \quad \forall (i,j) \in E \text{ such that } (i,j) \in p \\
&s_1 = 0\,, \\
&s_i \ge 0, \quad \forall i \in V \\
&\underline{\theta} \le \theta < \bar{\theta}\,.
\end{aligned} \tag{10}
$$

where $\underline{\theta} = L^{-1}(1)$, $\bar{\theta} = L^{-1}(0)$.

Thus, $Poss(p \text{ is critical}) = L(\theta)$ and $Nec(p \text{ is noncritical}) = 1 - L(\theta)$.

6 Conclusions

The project scheduling in networks with imprecise activities durations, which are represented by means of interval or fuzzy numbers, is a more realistic approach than the traditional approaches with crisp or stochastic activity durations. This approach partially destroys the usual critical path analysis, instead of being critical or not, the activities or paths that are necessarily critical, necessarily non-critical and possibly critical, thus the traditional critical path analysis is not robust. The proposed theorems determine whether a given path is necessarily critical, possibly critical or necessarily non-critical in interval-valued activity networks with generalized precedence relations which in addition to the common finish-start type of precedence constraints may specify arbitrary minimal and maximal time lags between the starting and completion times of the activities.

We have also shown how to calculate the path degree of possible criticality and necessary criticality based on linear programming which is true only for fuzzy activity and time lag durations determined by means of fuzzy number of the same *L-L* type.

References

[1] Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows. In: Nemhauser, G.L., Rinnooy Kan, A.H.G., Todd, M.J. (eds.) Handbooks in Operations Research and Management Science, pp. 258–263. Elsevier, Amsterdam (1989)

[2] Chanas, S., Dubois, D., Zielinski, P.: On the sure criticality of tasks in activity networks with imprecise durations. IEEE Trans. on Sys. Man, Cyber.-Part B 32, 393–407 (2002)

[3] Chanas, S., Zielinski, P.: Critical Path analysis in the network with fuzzy activity times. Fuzzy Set and Systems 122, 195–204 (2001)

[4] Chanas, S., Zielinski, P.: The computational complexity of the criticality of the problems in a network with interval activity times. European Journal of Operational Research 136(3), 541–550 (2002)

[5] De Reyck, B., Herroelen, W.: A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations. European Journal of Operational Research 111, 152–174 (1998)
[6] Dubois, D., Fargier, H., Fortin, J.: Computational methods for determining the latest starting times and floats of tasks in interval-valued networks. Journal of Intelligent Manufacturing 16, 407–421 (2005)
[7] Dubois, D., Prade, H.: Operations on fuzzy numbers. Int. J. Sys. Sci. 30, 613–626 (1978)
[8] Elmaghraby, S.E., Kamburowski, J.: The Analysis of Activity Network under Generalized Precedence Relations (GPRs). Management Science 38, 1245–1263 (1992)
[9] Fortin, J., Zielinski, P., Dubois, D., Fargier, H.: Interval analysis in scheduling. In: van Beek, P. (ed.) CP 2005. LNCS, vol. 3709, pp. 226–240. Springer, Heidelberg (2005)
[10] Kelley, J.E., Walker, M.R.: Critical path planning and scheduling. In: Eastern Joint Computer Conference, vol. 16, pp. 160–172 (1959)
[11] Lawler, E.L.: Combinatorial Optimization: Networks and Matroids. Holt, Rinehart, and Winston, New York (1976)
[12] Prade, H.: Using fuzzy sets theory in a scheduling problem: a case study. Fuzzy Sets and Systems 2, 153–165 (1979)
[13] Yakhchali, S.H., Fazel Zarandi, M.H., Turksen, I.B., Ghodsypour, S.H.: Possible criticality of paths in networks with imprecise durations and time lags. In: IEEE conference, NAFIPS, pp. 277–282 (2007)
[14] Yakhchali, S.H., Fazel Zarandi, M.H., Turksen, I.B., Ghodsypour, S.H.: Necessary criticality of paths in networks with imprecise durations and time lags. In: IEEE conference, NAFIPS, pp. 271–276 (2007)
[15] Yakhchali, S.H., Ghodsypour, S.H.: Hybrid genetic algorithms for computing the float of activities in networks with imprecise durations. In: IEEE international conference on fuzzy systems (FUZZ-IEEE), Hong Kong, pp. 1789–1794 (2008)
[16] Yakhchali, S.H., Ghodsypour, S.H.: Computing the latest starting times of activities in interval-valued networks with minimal time lags. European Journal of Operational Research (to appear)
[17] Yakhchali, S.H., Ghodsypour, S.H.: On latest starting times of activities in networks with imprecise durations and generalized precedence relations. Computers & Industrial Engineering (submitted)
[18] Yakhchali, S.H., Ghodsypour, S.H.: A Path Enumeration Approach for Temporal Analysis in Networks with imprecise durations and generalized precedence relations. Applied Soft Computing (submitted)
[19] Yakhchali, S.H., Ghodsypour, S.H.: On computing the maximal float of an activity in networks with interval durations and generalized precedence relations, working paper
[20] Zielinski, P.: On computing the latest starting times and floats of activities in a network with imprecise durations. Fuzzy Sets and Systems 159, 53–76 (2005)

Parallel Genetic Algorithm Approach to Automated Discovery of Hierarchical Production Rules

K.K. Bharadwaj and Saroj

Abstract. It is important to discover hierarchical decision rules from databases because much of the world's knowledge is best expressed in the form of hierarchies. Mining of decision rules at multiple concept levels leads to discovery of more informative and comprehensible knowledge. This paper proposes automated discovery of Hierarchical Production Rules (HPR) using a parallel genetic algorithm approach. A combination of degree of subsumption and coefficient of similarity has been used as a quantitative measure of hierarchical relationship among the classes. An island/deme GA is designed to evolve HPRs for the classes of the dataset being mined. The island model exploits control as well as data parallelism. The model is applied to a synthetic dataset on means of transport and results are presented.

1 Introduction

Knowledge Discovery in Databases (KDD) is defined as the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data [9]. Discovering meaningful and comprehensible knowledge has been of prime concern to the research community in the field of data mining. The Production Rules (PRs) *If*< Premise >*Then*< Decision > have been the most common representation for the discovery of decision rules. PRs ignore exceptions as noise and these are not capable of exhibiting variable precision logic. Moreover, PRs

K.K. Bharadwaj
School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India
e-mail: kbharadwaj@gmail.com

Saroj
Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar - 125001, Haryana, India
e-mail: ratnoo.saroj@gmail.com

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 327–336.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

discover knowledge at a single conceptual level fragmenting the knowledge into large number of rules. Mining of rules at multiple concept levels leads to discovery of more comprehensible and refined knowledge. Hierarchies allow us to manage the complexity of knowledge and to view the knowledge at different levels of details. Automatic generation of hierarchies as a post processing step has been effective to reduce the size of rule set, thereby, increasing the comprehensibility considerably [11]. In the recent past, on one hand researchers have focused on the discovery of exceptions on the other hand there has been various efforts to discover decision rules in hierarchical form [1][3][4][13][14]. This paper proposes automated discovery of Hierarchical Production Rules (HPRs) using a Parallel Genetic Algorithm (PGA) approach. The PGA discovers HPRs which keep track of the general and specific information of the decision class under consideration. A sequential evolutionary approach for the discovery of HPRs has already been given [4]. This approach is not sufficient for the voluminous datasets containing large number of classes where search space for the discovery of hierarchical rules is far too large to be handled by simple genetic algorithm. In this paper we have suggested a parallel GA which is more efficient and scalable. Moreover, this work modifies the fitness function used in [4] by including coefficient of similarity in addition to degree of subsumption to measure the goodness of hierarchies.

The rest of this paper is organized as follows. Section 2 introduces the necessary background details. Section 3 presents the design of the GA in detail. It explains the encoding, fitness function, the operators used and the scheme for parallelization. Section 4 illustrates the experimental set up and results. Section 5 gives the conclusions and scope for the future research.

2 Hierarchical Production Rules

Bharadwaj and Jain[5] have introduced a concept of Hierarchical Censored Production Rules (HCPRs) which is a Censored Production Rule (CPR) [12] augmented with specificity and generality information. A HCPR can be used to exhibit variable precision logic in reasoning such that both the certainty of belief in a conclusion and its specificity may be controlled by reasoning process[5][6]. A HCPR is represented as below:

Decision class *If* [condition(s)] *Unless*[censor(s)/Exception(s)]
Generality [general info]
Specificity [specific information]

This paper considers the discovery of HPRs having the generality and specificity parts of a HCPR, and leaving out the 'Unless' part. HCPRs allow only crisp hierarchy. In addition, we have included the possibility of hierarchies where the general and specific classes share quite some number of properties but neither is a proper subset of the other. A combination of degree of subsumption and coefficient of

similarity has been used to measure the hierarchical relationship among the classes. A typical HPR is given below:

D_1 *If* [Premise]

Generality[D_2(0.8, 0.66)] Specificity[D_3(0.8, 0.56), D_5(0.75, 0.71)]

The numbers in the braces show the degree of subsumption and coefficient of similarity respectively of the class D_1 with the general class D_2 and the specific classes D_3 and D_5.

2.1 Subsumption Relation

The defining properties of a class D_k are the distinct attributes with coverage greater than a user defined threshold (0.6 to 1.0). Let P_{D_k} and P_{D_s} be the sets of defining properties of classes D_k and D_s. The degree of subsumption between two classes D_k and D_s is defined below:

$$x = \left(\frac{|P_{D_k}(i) \wedge D_k|}{|D_k|}\right) \quad y = \left(\frac{|P_{D_s}(j) \wedge D_s|}{|D_s|}\right)$$

$$subsume\left(P_{D_k}(i), P_{D_s}(j)\right) = 1 \quad \text{if}\,(x \le y)\,and\,(i = j)$$

$$subsume\left(P_{D_k}(i), P_{D_s}(j)\right) = y \quad \text{if}\,(x > y)\,and\,(i = j)$$

$$subsume\left(P_{D_k}(i), P_{D_s}(j)\right) = 0 \quad \text{if}\,(i \neq j)$$

$$degreesub\,(D_k, D_s) = \sum_1^{|P_{D_k}|} \sum_1^{|P_{D_s}|} \frac{subsume\left(P_{D_k}(i),\, P_{D_s}(j)\right)}{|P_{D_k}|}$$

2.2 Coefficient of Similarity

A coefficient of similarity of attributes between two classes D_k and D_s is defined on the basis of a 2×2 contingency table as shown below:

Table 1 contingency table for P_{D_k} and P_{D_s}

$P_{D_k} \rightarrow$ $P_{D_s} \downarrow$	Observed	Not observed
Observed	a	b
Not Observed	c	d

$$\chi^2 = \frac{N(ad-bc)^2}{M}$$

$$N = a+b+c+d \quad M = (a+b)(c+d)(a+c)(b+d)$$

$$S = \sqrt{\frac{\chi^2}{N(k-1)}}$$

N is the total number of attributes present, k is the degree of freedom and in a 2×2 table, k=2; The value of S shall always be between 0 and 1. Higher the value of S, more is the similarity between the classes involved. A similar measure has been used by Tsumoto[14].

3 Design of Genetic Algorithm

GAs have been used in data mining for concept learning and rule discovery and have shown promising results[1][2][10][13]. Though GAs perform global search and cope better with attribute interactions, these take excessively long processing time for discovering rules from realistic volumes of data. Therefore, Parallel Genetic Algorithms offer a natural and cost effective choice for efficiency and scalability of rule discovery algorithms. Enrique Alba and Marco Tomassini, and Eric Centu Paz have written comprehensive survey papers on parallel GAs[7][8]. In this paper, we have used island/deme GA. Each deme runs an incremental GA in parallel to evolve HPRs for the classes assigned to it. A master processor collects the best HPRs evolved for each class. Each generation, the incremental GA selects two individuals by fitness proportionate selection and produces two new offspring. The newly produced individuals replace the two worst individuals in the population. This way the newly produced individuals are available to take part in evolution as soon as they are produced.

3.1 Encoding

The linear representation of the chromosome is divided into four parts. The decision part consists of the decision class under consideration (D_k) for which the HPR is being evolved. The 'Premise' part consists of conjunctions of attribute-value pairs. The 'Generality' part consists of the general class D_{kf} $(D_{kf} \neq D_k)$. The 'Specificity' part contains a maximum of three specific classes of the class D_k in a manner such that any $D_{si} \neq D_k$ and $D_{si} \neq D_{kf}$ and $D_{si} \neq D_{sj} (i, j = 1 \ldots 3; i \neq j)$. Only defining properties of the class D_k shall be considered for encoding the premise part of the individuals. This shall considerably reduce the search space for GA. A positional and fixed length chromosome, an alpha-numeric string with star (*) character as 'don't care' state symbol has been used. All the predicting values and classes are categorical. Our encoding follows the Michigan approach where each individual represents a single HPR. The linear structure for a chromosome class D_1 is shown below in Table 2. The chromosome would be decoded as:

Table 2 Structure for chromosome

Decision Part	If Part	Generality Part	Specificity Part
0	1**1*******1***	2	3*5

D_1 If [P1$\wedge$ P4 $\wedge$ P12] Generality[D_3] Specificity[D_4, D_6].

3.2 Fitness Function

Fitness function measures the goodness of the individual in the population. The fitness function for the discovery of HPRs is given below

Fitness for '*If-Then*' part
$F1 = confidencefactor \times coverage \times np$
$Confidencefactor = \frac{|P \wedge D_k|}{|P|}$
$Coverage = \frac{|P \wedge D_k|}{|D_k|}$ Coverage must be $\geq$ coverage threshold (0.8)
np = no. of properties in the premise
Fitness for '*Generality*' part
$F2 = degree - subsum\left(D_{kf}, D_k\right) \times similarity\left(D_{kf}, D_k\right)$
D_{kf} is the general class of D_k
Fitness for '*Specificity*' part
$F3 = \sum_1^3 degree - subsum\left(D_k, D_{si}\right) \times similarity\left(D_k, D_{si}\right)$
D_{si} is the specific class of the class D_k
$degree - sub(D_{kf}, D_k) \geq$ subsumption threshold(0.6)
$similarity(D_{kf}, D_k) \geq$ similarity thresholds (0.5)
$Fitness = F1 + F2 + F3$

3.3 GA Operators

Conventional GA operators of selection, crossover and mutation are used. We have used tree level and gene level one point crossover operators. A tree level crossover operator exchanges 'Premise' or 'Specificity' or 'Generality' parts of the two HPRs. Gene level crossover operator takes two HPRs and exchange of genes takes place within the properties of premise and/or 'Specificity' parts at random sites. Mutation performs insertion, deletion, value mutation and attribute mutation operations. After applying the GA operators to an individual, it must be ensured that the individual still meets the constraints of a HPR defined in the encoding section. Otherwise the individual is rejected. In addition to the conventional operators, an innovative post-processing fusion operator is used to fuse the related HPRs into a HPRs tree. The system can learn new concepts during the fusion as illustrated in the Sect. 4.

3.4 Parallel Genetic Algorithm Scheme

The parallel GA scheme suggested in this paper exploits control parallelism in evolving HPRs for different classes. We take n processors and these processors evolve autonomously and concurrently the HPRs for the classes assigned to them. Data is also partitioned into n equal size sets to exploit the data parallelism. In this scheme, migration operator does not move the copy of few individuals from one population to replace the worst members of another population. However, at every generation, migration operator moves the two newly produced offspring through all the processors in a ring topology to compute the '*If-Then*' part fitness $(F1)$. Each processor runs an incremental GA to evolve HPR from its local set of population and data .The 'If - Then' part fitness of newly produced individuals in populations is updated by having the individuals pass through all the processors in a ring topology.

Each processor passes its newly produced individuals to its right neighbour in the ring of processors. Each neighbour then updates the fitness measures of the individuals received, by accessing its local set of data and then it forwards the individuals to the next processor in the ring. This process continues until the individuals have returned to the processor they were originally produced at, with their fitness duly computed. Following the computation of '*If-Then*' fitness each processor computes the '*Generality*' and '*Specificity*' part fitness in parallel as given the Sect. 3.2. The detailed parallel genetic algorithm is given in appendix.

4 Experimental Setup and Results

We have evaluated our Deme Genetic Algorithm on a synthetic dataset on means of transport. The data set has 16 distinct values of attributes and 8 classes. The defining properties of the classes of the data set are shown in the Table 3. In all the cases, deme genetic algorithms has been run for 300 generations on populations of 30 individuals with crossover probability equals 0.7 and mutation probability equals 0.4. To implement the parallel genetic algorithms GALIB245 and Parallel Virtual Machine (PVM) have been used. Two different dataset sizes of 4000 and 8000 instances have been considered. The HPRs evolved for different classes are presented in Table 4.

Further conceptual hierarchies have been generated by fusing the related HPRs. As 'car', 'motor bike' and 'train' classes subsume each other with same degree of subsumption (0.8) and these classes are similar (0.71) to each other. Therefore, a new class 'land transport' can be conceptualized. The new class shall have common properties from all the three classes and subsume each of its specific class with a degree of subsumption one. The coefficient of similarity between the new class and its specific classes can be recomputed. The classes can only be merged if the threshold criteria for the degree of subsumption and similarity coefficient are met. In our dataset, no further fusion of HPRs is possible. Therefore the classes 'airplane' and 'space rocket' do not form the part of hierarchy. The hierarchies generated are shown in Fig. 1.

The parallel scheme has achieved super linear speed gains. The speed-ups due to parallelism at two dataset sizes are shown in Table 5. The speed gains originate from the following two sources: 1. Distribution of the classes among the processors: In sequential GA, a single processor has to evolve HPRs for the eight classes of 'Transport' dataset by running the incremental GA eight times in succession. When we have more processors, classes are evenly distributed among the processors using control parallelism. For example when we have eight processors, each processor evolves the HPRs for a single class only. 2. Data parallelism: In case of dataset of size 4000, the speed gain is super linear when number of processors involved is 2 and 4. When number of processors is eight, the speed gain goes below linearity. This is due to the increase in communication cost as individuals of the population migrate to the various processors to compute the fitness. At a dataset size of 8000, the speed gains are amplified. The speed gain remains super linear even when

Table 3 Frequency matrix for Means of Transport

Attributes		Ship	Car	Airplane	Motor-bike	Submarine	Train	Rocket	Boat
Driver	:P1	1	1	1	1	1	1	0	1
Solid booster	:P2	0	0	0	0	0	0	1	0
Railway tracks	:P3	0	0	0	0	0	1	0	0
Wings	:P4	0	0	1	0	0	0	0	0
Moves on land	:P5	0	1	0	1	0	1	0	0
Orbiter	:P6	0	0	0	0	0	0	1	0
Fuel	:P7	1	1	1	1	1	1	1	0
Swimming	:P8	1	0	0	0	1	0	0	1
Anchor based	:P9	1	0	0	0	0	0	0	1
Fly	:P10	0	0	1	0	0	0	1	0
Move under sea	:P11	0	0	0	0	1	0	0	0
Engine	:P12	1	1	1	1	1	1	1	0
Manual force	:P13	0	0	0	0	0	0	0	1
wheel=0	:P14	1	0	0	0	1	0	1	1
Wheel=2	:P15	0	0	0	1	0	0	0	0
Wheel=4	:P16	0	1	0	0	0	0	0	0

Table 4 Hierarchical Production Rules for the 'Means of Transport' dataset

No.	HP Rs	F1	Sub	Si mi	F2	Sub	Si mi	F3	Fitness
1.	Ship *If* [Driver ∧ Fuel ∧ Swim ∧ Anchor based ∧ Engine ∧ Wheel=0] *G*[boat] *S*[submarine]	6.0	0.8	0.57	0.456	0.75	0.83	0.62	7.056
2	Car *If* [Driver ∧ Moves-on-land ∧ Fuel ∧ Engine ∧ Wheel=4] *G*[Motor bike] *S*[Train]	5.0	0.8	0.71	0.57	0.8	0.71	0.57	6.14
3	Airplane *If* [Driver∧ Fuel ∧ Engine ∧ Fly ∧ Wings] $G\tilde{[]}$ $S\tilde{[]}$	5.0	0	0	0	0	0	0	5.0
4	Motorbike *If* [Driver ∧Moves on land ∧ Fuel∧ Engine ∧ Wheel=2] *G*[Car] *S*[Train]	5.0	0.8	0.71	0.57	0.8	0.71	0.57	6.14
5	Submarine *If* [Driver∧ Fuel∧ Swim∧ Engine ∧ Wheel=0 ∧ Move under sea] *G*[Ship] $S\tilde{[]}$	6.0	0.83	0.75	0.62	0	0	0	6.62
6	Train *If* [Driver ∧ Moves on land ∧ Fuel ∧ Engine ∧ Track] *G*[Motor bike] *S*[Car]	5.0	0.8	0.71	0.57	0.8	0.71	0.57	6.14
7	Space-rocket *If* [Fuel ∧ Engine ∧ Fly ∧ Orbital ∧ Solid booster ∧wheels =0] $G\tilde{[]}$ $S\tilde{[]}$	6.0	0	0	0	0	0	0	6.0
8	Boat *If* [Driver ∧ Swim ∧ Anchor-based ∧ Manual force ∧Wheel=0] $G\tilde{[]}$ *S*[Ship]	5.0	0	0	0	0.8	0.57	0.456	5.456

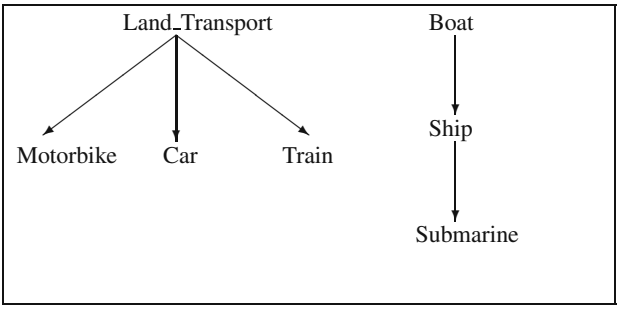


Fig. 1 Overall hierarchy for the 'Means of Transport' dataset

Table 5 Speed-ups due to parallelism

Data Size	Processors(P_n)	Time(Seconds)	Speed Gain
4000	1	51	-
	2	22	2.31
	4	12	4.25
	8	7	7.28
8000	1	92	-
	2	37	2.48
	4	17	5.41
	8	9	10.22

number of processors is increased to eight. This clearly shows that the larger dataset is able to take more advantage of data parallelism. In case of larger dataset size, the time to evaluate fitness function is large enough to dominate the communication overhead of the migrating individuals to implement data parallelism.

5 Conclusions and Future Direction

A parallel genetic algorithm approach is designed to discover HPRs. A combination of the concepts of subsumption and similarity coefficient has been used to measure the inheritance relationships quantitatively between the general and specific classes. The approach is employed to a synthetic data set on means of transport. The knowledge discovered at multiple conceptual levels is concise and comprehensible. A reasoning system using HPRs as its underlying knowledge structure can exhibit variable precision logic regarding specificity. Given the time constraint, the system can come up with general responses otherwise it can give more specific answers. Our system can learn new concepts while fusing togather the related

HPRs. The relationship of generality and specificity in HPRs shall be helpful for forward and backward chaining of a reasoning system. The parallel GA has been effective and has shown super linear speed gains. It would be interesting to see how the approach performs on real world larger datasets with many more classes. The most important future research is to discover Hierarchical Censored Production Rules.

References

1. Aguilar-Ruiz, J.S., Requelme, J.C., Toro, M.: Evolutionary Learning of Hierarchical Decision Rules. IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetic. 33, 324–331 (2003)
2. Araujo, D.L.A., Lopes, H.S., Freitas, A.A.: A Parallel Genetic Algorithm for Rule Discovery in Large Databases. In: Proc. IEEE Systems, Man and Cybernetics Conference, Tokyo, vol. 3, pp. 940–945 (1999)
3. Bharadwaj, K.K., Al-Maqaleh, B.M.: Evolutionary Approach for Automated Discovery of Censored Production Rules. Enformatika 10, 147–152 (2005)
4. Bharadwaj, K.K., Al-Maqaleh, B.M.: Evolutionary Approach for Automated Discovery of Censored Production Rules with Fuzzy Hierarchy. In: International Multi-conference of Engineers and Computer Scientists, Hong kong, pp. 716–721 (2007)
5. Bharadwaj, K.K., Jain, N.K.: Hierarchical Censored Production Rules (HCPRs) System. Data Mining and Knowledge Engg. 8, 19–34 (1992)
6. Bharadwaj, K.K., Jain, N.K.: Some Learning Techniques in Hierarchical Censored Production Rules (HCPRs) System. International Journal of Intelligent Systems 13, 319–344 (1997)
7. Centu-Paz, E.: A Survey of Parallel Genetic Algorithm. Calculateurs Paralleles, Reseaux et Systems, Repatis. 10, 141–147 (1998)
8. Enrique, A., Tomassini, M.: Parallel and Evolutionary Algorithms. IEEE Transactions on Evolutionary Computation 6, 443–462 (2002)
9. Fayyad, U.M., Shapiro, G.P., Smyth, P.: The KDD Process for Extracting Useful Knowledge from Volumes of Data. Communication of ACM 39, 27–34 (1996)
10. Freitas, A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer, Berlin (2002)
11. Kandwal, R., Bharadwaj, K.K.: Cumulative Growth of Production Rules with Fuzzy Hierarchy (PRFH). In: Proceeding of the IASTED International Conference on Artificial Intelligence and Soft Computing (ASC 2006), Palma De Mallorca, Spain, pp. 40–45. ACTA Press (2006)
12. Michalski, R.S., Winston, P.H.: Variable Precision Logic. Artificial Intelligence 29, 121–146 (1986)
13. Saroj, S., Bharadwaj, K.K.: A Parallel Genetic Algorithm Approach to Automated Discovery of Censored Production Rules. In: Proceedings of IASTED International Conference on Artificial Intelligence and Applications (AIA 2007), Innsbruck, Austria, pp. 435–441. ACTA press (2007)
14. Tsumoto, S.: Automated Extraction of Hierarchical Decision Rules from Clinical Databases using Rough Set Model. Expert System Applications 24, 189–197 (2003)

Appendix - I

A Parallel genetic algorithm to evolve HPRs is given below:

```
Procedure PGA HPRs
var inputs
S: Dataset;
nc: no. of classes in the dataset S to be mined
var outputs
H: A set of HPR trees, HPRs and PRs;
np: integer // no. of processors
p: population; no-of-generations:integer
no-of-children: integer;
begin
    Choose np such that: nc = l*np; /*l: integer */
    Divide the dataset S equally on to np processors;
    Assign the classes equally among the processors
    Designate one processor as the master processor;
    Fori=1 to np execute in parallel
        begin
        Initialize random population p(i) of HPRs
        on i^th processors for the i^th class;
        Compute-fitness(p(i)); /* as given in sec. 3.4 */
        Forj= 1 to no-of-generations
            begin
                Select the two individuals with roulette wheel
                Apply GA operators to the individuals to
                produce two new off-springs;
                Compute the fitness of new off-springs;
                Replace the two worst individuals of the
                population with the two new off-springs;
            end
        Send the best HPRs from each processor to the master processor;
        end
Master processor applies fusion to merge the related HPRs into HPR trees;
/* The output is a set H of rules containing HPR trees, isolated HPRs and PRs */
end
stop.
```

Two Hybrid Genetic Algorithms for Solving the Super-Peer Selection Problem*

Jozef Kratica, Jelena Kojić, Dušan Tošić, Vladimir Filipović, and Djordje Dugošija

Abstract. The problem that we will address here is the Super-Peer Selection Problem (SPSP). Two hybrid genetic algorithm (HGA) approaches are proposed for solving this NP-hard problem. The new encoding schemes are implemented with appropriate objective functions. Both approaches keep the feasibility of individuals by using specific representation and modified genetic operators. The numerical experiments were carried out on the standard data set known from the literature. The results of this test show that in 6 out of 12 cases HGAs outreached best known solutions so far, and that our methods are competitive with other heuristics.

1 Introduction

During the past 20 years, computer networks have been rapidly expanding because of users' need to connect to other computers via local or global computer networks. Some network strategies are based on client/server architecture, while the others are Peer-to-Peer (P2P) systems. P2P networks are fully decentralized and the advantages of this approach are self-organizing and fault-tolerant behavior. Failure in a single node usually does not affect the entire network at once. However,

Jozef Kratica
Mathematical Institute, Serbian Academy of Sciences and Arts, Kneza Mihajla 36/III,
11 001 Belgrade, Serbia
e-mail: jkratica@mi.sanu.ac.yu

Jelena Kojić · Dušan Tošić · Vladimir Filipović · Djordje Dugošija
University of Belgrade, Faculty of Mathematics, Studentski trg 16/IV,
11 000 Belgrade, Serbia
e-mail: k_jelena@yubc.net, tdusan@mi.sanu.ac.yu,
vladaf@matf.bg.ac.yu, dugosija@matf.bg.ac.yu

* This research was partially supported by Serbian Ministry of Science under the grant no. 144007.

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 337–346.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

excessive growth of network communication may affect the scalability of such networks. Namely, in the case of larger networks, communication times tend to increase and the load put on every node grows significantly. This problem can be solved by the introduction of super-peers. Super-peers are peers that act as servers for a number of attached common peers, while at the same time, forming a network of nodes equal among themselves. In this super-peer P2P network, each common peer is assigned to exactly one super-peer, which represents its only link to the rest of the network. Obviously, all communications between different peers must be routed via at least one super-peer. The presentation of all relevant information about Peer-to-Peer networks is out of the scope of this paper and details can be found in [6, 14].

Although there are many papers about P2P networks, we have found only two papers that deal with the selection of super-peers in order to minimize overall network communication. One of the papers [12] proves that the Super-Peer Selection Problem (SPSP) is NP-hard. The other paper [13]] suggests a solution to this problem. A new metaheuristic based on evolutionary techniques combined with local search is presented. The proposed metaheuristic is tested on real-world instances based on actual internet distance measurements. Significant savings in total communication costs are demonstrated for all instances in contrast to a case without super-peer topology.

2 Mathematical Representation

We used an integer programming representation of SPSP similar to the one in [13]. Since the number of nodes assigned to a super-peer network is limited, we incorporated this into the model. We defined n as the number of nodes, p as the number of super-peer nodes, and d_{ij} as the distance between nodes i and j. Let $x_{ij} \in \{0,1\}$ have the value 1 if node i is allocated to a super-peer j and 0 otherwise. The condition $x_{kk} = 1$ implies that the node k is a super-peer.

The problem can be expressed thus:

$$min \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} (d_{ik} + d_{kl} + d_{lj}) x_{ik} x_{jl} \tag{1}$$

subject to:

$$\sum_{k=1}^{n} x_{kk} = p \tag{2}$$

$$\sum_{k=1}^{n} x_{ik} = 1 \quad \forall i = 1, ..., n \tag{3}$$

$$\frac{p}{2} \cdot x_{kk} \leq \sum_{i=1}^{n} x_{ik} \leq 2p \cdot x_{kk} \quad \forall k = 1, ..., n \tag{4}$$

$$x_{ik} \in \{0,1\} \quad \forall i, k = 1, ..., n \tag{5}$$

The objective function (1) minimizes the overall costs. The constraint (2) ensures that exactly p super-peers are chosen, while the constraint (3) guarantees a single super-peer allocation for each node. The constraint (4) limits the number of nodes assigned to each super-peer k to between $p/2$ and $2p$. The constraint (4) implies also that if k is a non-super-peer node ($x_{kk} = 0$), then $x_{ik} = 0$ holds for every i . It is easy to see that the flow goes only via super-peer nodes, thus preventing direct transmission between other nodes. By the constraint (5) we prevent super-peers being allocated to other nodes.

3 Proposed Hybrid GA Methods

GAs are stochastic search techniques imitating some spontaneous optimization processes in the natural selection and reproduction. At each iteration (generation), a GA manipulates a set (population) of encoded solutions (individuals), starting from either randomly or heuristically generated one. Individuals from a current population are evaluated using a fitness function to determine their qualities. Good individuals are selected to produce new ones (offspring), applying operators inspired from genetics (crossover and mutation), and they replace some of the individuals from the current population. A detailed description of GAs is out of this paper's scope and can be found in [9, 8, 10]. Extensive computational experience of various optimization problems shows that a GA often produces high quality solutions within a reasonable time. Some of the recent applications related to this problem are [4, 5, 11, 7].

3.1 Description of HGA1 Representation

The genetic code of each individual consists of n genes, each referring to one network node. The first bit in each gene takes the value 1 if the current node is a super-peer, 0 if it is not. Considering these bit values, we form an array of opened super-peers. If a node is a super-peer the remaining bits of the gene are ignored, while if it is not a super-peer node, the remaining part of the gene r_{nsp} refers to the super-peer assigned to the current non-super-peer node nsp. Details of this assignment will be explained later as the nearest neighbour ordering.

3.2 Description of HGA2 Representation

In this representation of the GA, the genetic code of an individual consists of two segments. The first segment is a string of length p, where the digits (genes) take values from the set $\{0, 1, ..., n-1\}$. Each digit in this segment shows which nodes are set as super-peers. The duplication of a super-peer index is resolved in the following way: if an index repeats in a genetic code, we replace it by the next previously unused one. If there are no such indices, we use the previous index that was not already

taken. Since p is smaller than n, we will always find a "free" index to replace the duplicated one. This approach ensures that exactly p distinct super-peer indices are obtained from a genetic code.

The second segment in the genetic code has exactly $n - p$ genes, where each gene r_{nsp} corresponds to the non-super-peer node nsp. The gene value refers to the super-peer assigned to the current node, also applying the nearest neighbor ordering.

3.3 Objective Function

For each non-super-peer node nsp the nearest neighbor ordering is applied in several steps:

Step 1. Super-peers that reached $2p$ assignments are discarded from consideration;
Step 2. The remaining super-peers are arranged in non-decreasing order of their distances from a particular node;
Step 3. The super-peer with the index r_{nsp} in the arranged order is assigned for the current non-super-peer node nsp.

This procedure guarantees that every super-peer node has at most $2p$ node assignments. If some super-peers have less than $p/2$ node assignments, we look for nodes that are assigned to super-peers with more than $p/2$ assignments and choose the one with the best change of overall cost. This is repeated until all super-peers have at least $p/2$ node assignments.

The super-peers that are closer to non-super-peer nodes appear often in the optimal solution, while the far away super-peers are rare. For this reason, we directed our search to "closer" super-peers, while the "distant" ones were considered of small probability. As a result, the values of r_{nsp} had to be relatively small, otherwise the nearest neighbor ordering would become a classical GA search. If all $r_{nsp} = 0$, this would turn into a greedy search.

On this solution, we applied local search procedures as proposed in [13]. There were three different neighborhoods: replacing the super-peer, swapping two peers and reassigning a peer to another super-peer. Since their running times were very time consuming, we used these local search procedures occasionally, in contrast to [13], where all three procedures were used on each individual in every generation. In the first 20 generations, we did not use the local search at all. Later, if the best individual was changed, we improved it, but if the best individual was not changed in the last 5 generations, we randomly chose some individual and applied all three local search procedures to it.

3.4 Selection

The selection operator, which chooses parent individuals for producing offspring in the next generation, is an improved tournament selection operator known as the fine-grained tournament selection (see [2]). This operator uses a real

(rational) parameter F_{tour} denoting the desired average tournament size. Two types of tournament were performed: the first type was held k_1 times on $\lfloor F_{tour} \rfloor$ individuals, while the second type was applied k_2 times with $\lceil F_{tour} \rceil$ individuals participating, so $F_{tour} \approx \frac{k_1 \cdot \lfloor F_{tour} \rfloor + k_2 \cdot \lceil F_{tour} \rceil}{N_{nnel}}$, where N_{pop} and N_{elite} was the overall number of individuals and number of elitist individuals in the population, and $N_{nnel} = N_{pop} - N_{elite}$.

In our implementation $F_{tour} = 5.4$ and, corresponding values k_1 and k_2 for $N_{nnel} = 50$ non-elitist individuals were 30 and 20, respectively.

3.5 Crossover and Mutation

After a pair of parents was selected, the crossover operator was applied to them producing two offspring. The crossover operator in HGA1 simultaneously traced genetic codes of the parents from right to left, searching the gene position i where the parent1 had "1-bit" and the parent2, "0-bit" on the first bit position of these genes. The individuals then exchanged whole genes on the gene position i. The corresponding process was simultaneously performed on gene position j, starting from the left side of the parents' genetic codes. The described process was repeated until $j \geq i$.

Note that the number of located super-peers in both offspring remained unchanged compared to their parents. Since the parents were correct, their offspring were correct, too. The implemented crossover operator was applied with the rate ($p_{cross} = 0.85$).

In HGA2 each parent's genetic code consisted of two segments of different nature. We applied a double one-point crossover: in each segment of the parents' genetic codes, one crossover point was randomly chosen and the genes were exchanged after the chosen position.

The offspring generated by a crossover operator were subject to mutation. The mutation operator, when applied, changed a randomly selected gene in the genetic code. During the GA execution, it is possible that all individuals in the population have the same gene in a certain position. These genes are called frozen. If the number of frozen genes is significant, the search space becomes much smaller, and the possibility of premature convergence rapidly increases. For this reason, basic mutation rates are increased, but only for the frozen genes. The basic mutation rates are:

- $0.2/n$ for the bit in the first position.
- $0.05/n$ for the bit in the second position. Next bits in the gene have repeatedly two times smaller mutation rate ($0.025/n$, ...).

When compared to the basic mutation rates, frozen bits are mutated by:

- 2.5 times higher rate ($0.5/n$ instead of $0.2/n$) if they are in the first position in the gene.
- 1.5 times higher rate ($0.075/n$, $0.0375/n$, ...) otherwise.

In HGA1, the previous process could not guarantee the feasibility of individuals, so we counted and compared the number of mutated ones and zeros in the first bits of genes in each individual. In cases where these numbers were not equal, it was necessary to mutate additional leading bits of genes. Equalizing the number of mutated ones and zeros in leading positions, the mutation operator preserved exactly p super-peers and preserved the feasibility of the mutated individuals.

3.6 Caching GA

The main purpose of caching was to avoid the calculation of objective values for individuals that reappeared during the a GA run. The evaluated objective values were stored in a hash-queue data structure, which was created using the Least Recently Used (LRU) caching strategy. When the same code was obtained again, its objective value was taken from the hash-queue table, instead of calculating its objective function. The implemented caching technique improved the GA running time (see [3]). The number of cached objective values in the hash-queue table was limited to N_{cache} = 5000 in our HGA implementations.

3.7 Other GA Aspects

The population numbered 150 individuals, but for the applied encodings, random population in the first generation was not appropriate. These encodings, based on the nearest neighbor ordering, as described above, showed that numbers r_{nsp} must be small. If pr_i, $i = 0, ..., p-1$ is the probability that $r_{nsp} = i$, for every non-super-peer node must hold $\sum_{i=1}^{p-1} pr_i = 1$ and $pr_0 \geq pr_1 \geq ... \geq pr_{p-1}$. The appropriate model was the geometric progression for $pr_0 = \frac{1-q}{1-q^p}$, where q was the common ratio. In our case, q was 0.4. The initial population was generated according to these probabilities.

Steady-state generations were replaced by applying elitist strategy. In this replacement scheme, only N_{nonel} = 50 individuals were replaced in every generation, while the best N_{elite} = 100 individuals directly passed into the next generation preserving highly fitted genes. The elite individuals did not need recalculation of objective values, since each of them was evaluated in one of the previous generations.

Duplicated individuals were removed from each generation. Their fitness values were set to zero, so that the selection operator prevented them from entering the next generation. This was a highly effective method of preserving the diversity of genetic material and keeping the algorithm away from premature convergence. Individuals with the same objective function but different genetic codes can in some cases dominate the population. If their codes are similar, the GA can lead to a local optimum. For that reason, it is useful to limit their appearance to some constant. In this GA application this constant was set to 40.

4 Computational Results

The GA tests were performed on an Intel 2.66 GHz with 4 GB RAM, under Linux (Knoppix 5.3.1) operating system. The GA stopping criterion was the maximum number of generations equal to 5,000 or at most 2,000 generations without an improvement of the objective value. Testing of both HGA approaches was performed on real world instances from [13] based on node-to-node delay information from PlanetLab [1], a world-wide platform for performing Internet measurements. The dimension of these instances varied from $n = 70$ to $n = 419$ nodes and $p \approx \sqrt{(n)}$. Note that for these instances triangle inequalities are frequently violated due to different routing policies and missing links most likely due to firewalls (for more details see [1, 13]).

Table 1 Characteristic of SPSP instances

Inst	*n*	*p*	*Best known solution*	
01-2005	127	12	2927946	*new*
02-2005	321	19	18596988	*new*
03-2005	324	18	20642064	*new*
04-2005	70	9	739954	
05-2005	374	20	25717036	
06-2005	365	20	22311442	*new*
07-2005	380	20	31042374	*new*
08-2005	402	21	30965218	
09-2005	419	21	33014358	*new*
10-2005	414	21	32922594	
11-2005	407	21	27902552	
12-2005	414	21	28516682	

Table 1 contains current best known solutions, based on results from [13] and new results from HGA1 and HGA. The table is organized as follows:

- the first column contains the instance name;
- in the second and the third columns are the number of nodes, n, and the number of super-peers, p, respectively;
- current best known solutions are given in the last column. Solutions achieved by HGAs are marked with *new*.

The HGA1 and HGA2 were run 30 times for each instance and the results were summarized in Tables 2 and 3, respectively. The tables are organized as follows:

- the first two columns contain the instance name and the best known solution from Table 1;
- the third column, named GA_{best}, contains the best HGA solutions. The solutions that are equal to the best known are marked *best*;
- the average running time (t) used to reach the final GA solution for the first time is given in the fourth column, while the fifth and sixth columns (t_{tot} and *gen*)

Table 2 Results of the HGA1

Inst	$Best_{sol}$	GA_{best}	t (sec)	t_{tot} (sec)	*gen*	*agap* (%)	σ (%)	*eval*	*cache* (%)
01-2005	2927946	*best*	50.8	73.7	4675	0.126	0.287	128152	45.2
02-2005	18596988	18621460	495.7	660.6	4642	0.417	0.283	146473	37.0
03-2005	20642064	20650488	590.4	730.7	4785	1.733	0.757	137183	42.6
04-2005	739954	*best*	7.5	13.2	3851	0.609	1.480	89222	53.1
05-2005	25717036	25727144	929.7	1098.1	4875	0.285	0.328	148494	39.1
06-2005	22311442	*best*	605.0	854.8	4527	0.175	0.125	132700	41.4
07-2005	31042374	31082632	784.4	992.6	4840	0.332	0.237	136918	43.4
08-2005	30965218	30971996	902.5	1145.0	4793	0.634	0.292	154109	35.7
09-2005	33014358	*best*	924.1	1233.8	4653	0.917	0.441	152024	34.8
10-2005	32922594	32966074	1017.5	1237.6	4778	0.820	0.520	154744	35.1
11-2005	27902552	27942502	909.7	1198.1	4783	1.208	0.662	160861	32.9
12-2005	28516682	28561004	1009.7	1300.8	4849	0.564	0.371	152419	37.2

Table 3 Results of the HGA2

Inst	$Best_{sol}$	GA_{best}	t (sec)	t_{tot} (sec)	*gen*	*agap* (%)	σ (%)	*eval*	*cache* (%)
01-2005	2927946	*best*	31.0	52.9	4111	1.386	0.310	88090	57.1
02-2005	18596988	*best*	478.9	597.4	4784	0.457	0.202	106383	55.5
03-2005	20642064	*best*	553.3	675.6	4821	1.437	1.023	107753	55.3
04-2005	739954	*best*	3.6	10.1	2984	0.246	0.068	76212	49.1
05-2005	25717036	25722412	762.0	942.9	4773	1.669	0.546	108867	54.5
06-2005	22311442	22326972	657.3	822.0	4865	0.385	0.232	106119	56.3
07-2005	31042374	*best*	716.0	885.1	4810	4.955	0.995	104526	56.6
08-2005	30965218	31050464	856.9	1053.2	4763	0.542	0.280	111634	53.2
09-2005	33014358	33188802	927.1	1094.5	4783	0.834	0.351	105898	55.7
10-2005	32922594	33054968	959.7	1121.7	4921	1.257	0.444	109063	55.7
11-2005	27902552	27956910	874.0	1042.9	4795	1.707	0.707	113642	52.4
12-2005	28516682	28536494	1014.1	1191.7	4916	0.441	0.269	106272	56.7

show the average total running time and the average number of generations for finishing GA, respectively. Note that running time t_{tot} includes t;

- the seventh and eighth columns (*agap* and σ) contain information on average solution quality: *agap* is a percentage gap defined as $agap = \frac{1}{20}\sum_{i=1}^{20} gap_i$, where $gap_i = 100 * \frac{GA_i - Best_{sol}}{Best_{sol}}$ and GA_i represents the GA solution obtained in the i-th run, while σ is the standard deviation of gap_i, $i = 1, 2, ..., 20$, obtained by formula $\sigma = \sqrt{\frac{1}{20}\sum_{i=1}^{20}(gap_i - agap)^2}$.
- in the last two columns, *eval* represents the average number of objective function evaluations, while *cache* displays savings (in percentages) achieved by using the caching technique.

5 Conclusions

We have described two evolutionary metaheuristics for solving SPSP. For each method, two-segment encoding of individuals and appropriate objective functions were used. Arranging super-peers in non-decreasing order of their distances from each node directed GA to promising search regions. The initial population was generated to be feasible and genetic operators adapted to SPSP were designed and implemented. Improving heuristic based on local search additionally improved solutions given by genetic algorithms. Genetic operators preserved the feasibility of solutions, so the incorrect individuals did not appear throughout all generations. We have used the idea of frozen bits to increase the diversity of the genetic material. The caching technique additionally improved the computational performance of both HGAs.

The results clearly demonstrated usefulness of our hybrid GA approaches with new best solutions for six of the SPSP instances. Hence, our future work could concentrate on the parallelization of the HGAs and their hybridization with exact methods. Based on the results, we believe that our HGA approaches have a potential as useful metaheuristics for solving other similar problems that arise in Peer-to-Peer network communications.

References

1. Banerjee, S., Griffin, T.G., Pias, M.: The Interdomain Connectivity of PlanetLab Nodes. In: Barakat, C., Pratt, I. (eds.) PAM 2004. LNCS, vol. 3015, pp. 73–82. Springer, Heidelberg (2004)
2. Filipović, V.: Fine-grained tournament selection operator in genetic algorithms. Computing and Informatics 22, 143–161 (2003)
3. Kratica, J.: Improving performances of the genetic algorithm by caching. Computers and Artificial Intelligence 18, 271–283 (1999)
4. Kratica, J., Stanimirović, Z.: Solving the Uncapacitated Multiple Allocation p-Hub Center Problem by Genetic Algorithm. Asia-Pacific Journal of Operational Research 23, 425–438 (2006)
5. Kratica, J., Stanimirović, Z., Tošić, D., Filipović, V.: Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem. European Journal of Operational Research 182, 15–28 (2007)
6. Li, D., Xiao, N., Lu, X.: Topology and resource discovery in Peer-to-Peer overlay networks. In: Jin, H., Pan, Y., Xiao, N., Sun, J. (eds.) GCC 2004. LNCS, vol. 3252, pp. 221–228. Springer, Heidelberg (2004)
7. Marić, M.: An efficient genetic algorithm for solving the multi-level uncapacitated facility location problem. Computing and Informatics (in press)
8. Merz, P.: Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies. PhD thesis, Department of Electrical Engineering and Computer Science, University of Siegen, Germany (2000)
9. Merz, P., Freisleben, B.: Fitness landscapes and memetic algorithm design. In: Corne, D., Dorigo, M., Glover, F. (eds.) New ideas in optimization, pp. 245–260. McGraw-Hill, London (1999)

10. Mitchell, M.: An introduction to genetic algorithms. MIT Press, Cambridge (1999)
11. Stanimirović, Z.: A genetic algorithm approach for the capacitated single allocation p-hub median problem. Computing and Informatics 27 (in press)
12. Wolf, S.: On the complexity of the uncapacitated single allocation p-hub median problem with equal weights. Technical report, University of Kaiserslautern, Distributed Algorithms Group, Internal Report No. 363/07 (July 2007)
13. Wolf, S., Merz, P.: Evolutionary local search for the super-peer selection problem and the *p*-hub median problem. In: Bartz-Beielstein, T., Blesa Aguilera, M.J., Blum, C., Naujoks, B., Roli, A., Rudolph, G., Sampels, M. (eds.) HM 2007. LNCS, vol. 4771, pp. 1–15. Springer, Heidelberg (2007)
14. Yang, B., Garcia-Molina, H.: Designing a super-peer network. In: Proceedings of the 19th International Conference on Data Engineering, Bangalore, India, pp. 49–62 (2003)

A Genetic Algorithm for the Constrained Coverage Problem

Mansoor Davoodi, Ali Mohades, and Jafar Rezaei

Abstract. The coverage problem is one of the most important types of the facility location problems, which belongs in the *NP*-hard problems. In this paper, we present a genetic algorithm for solving the constrained coverage problem in continuous space. The genetic operators are novel operators and specially designed to solve the coverage problem. The new algorithm has a high convergence rate and finds the global optimum by a high probability. The algorithm is tested by several benchmark problems, the results of which demonstrate the power of algorithm.

1 Introduction

Coverage problems are one of the most practical subjects in the sensor networks and facility location that have been studied in the past years. In general, the coverage problem is put forward as the *k*-coverage form and has been discussed on the continuous or discrete, constrained or unconstrained, and weighted or un-weighted space. In the discrete space, there are n targets (or demand points) and the goal is to cover them by sensors that have the certain effective sense radius which we denote by *ESR*. Any sensor can be displayed as the center of a circle with radius *ESR* (denoted by *ESR*-sensor). By such representation, any demand point is covered by a sensor if and only if the distance from the sensor is equal to or less than *ESR*. In the k-coverage problems, each demand point is covered by at least *k* different *ESR*-sensors, where the number of sensors is minimized. In general, this problem is *NP*-hard [1], and for this reason many heuristic algorithms have been presented for solving it (cf. [2 - 6]).

In the simplest case, when k equals one, the *1*-coverage problem is put forward, which is called the coverage problem here for simplicity. In this problem, the goal

Mansoor Davoodi · Ali Mohades
Laboratory of Algorithms and Computational Geometry, Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran
e-mail: mansoorcom81@yahoo.com, {mdmonfared,Mohades}@aut.ac.ir

Jafar Rezaei
Faculty of Technology, Policy and Management, Delft University of Technology, The Netherlands
e-mail: j.rezaei@tudelft.nl

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 347–356.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

is to find the minimum number of ESR-sensors such that they cover all demand points. In [7] continuous space has been studied, where demand points constitute an infinite set and are modeled by polygons. Generally, the continuous space coverage problem falls into complete and approximate coverage and can be solved using gridding or partitioning approaches [7, 8]. In the constrained versions, which are one step closer to real world problems, there are some constraints on the position of sensors. These constraints, usually shown by polygons, represent the real constraints like highways, lakes, etc.

One of the other types of facility location problems, of which various versions have been studied in the recent years, is the *p*-center problem [9-11], that is applied in emergency facility location. In this problem, which is partly similar to the coverage problem, the goal is to find exactly p circles that cover all demand points and minimize the maximum radius of the circles. Similarly in this problem if the demand points have different preferred levels, a specific weight is assigned to any of them, and if the space is continuous the density is considered non-uniform [12, 13]. Recently, in [14], a heuristic algorithm based on Voronoi diagram has been used to solve the constrained *p*-center problem in continuous space, and applied to constrained space in a real world problem for locating warning sirens.

In addition to the applications of the coverage problem in sensor networks and wireless antennas [15], this problem is also applied in military applications, object tracking, data gathering, etc. [16-17]. Regarding to the widespread applications of the coverage problems, the goal of this paper is to present a Genetic Algorithm (*GA*) for solving the constrained continuous space *p*-center problem. Satisfying constraints, *GA* uses a similar way that is applied in [14] for constrained p-center problem and explained in section 3. Also in *GA*, more standard operators (selection, crossover and mutations [18]) are used as transfer operators that transfer each center to the center of Minimum Covering Circle (*MCC*) of a set of demand points. The proposed crossover and mutation operators are different from standard ones that are usually applied in evolutionary optimization problems.

Objects which are used on the body of new approach are presented in the next section. Section 3 explains the constrained coverage problem. Section 4 presents a genetic algorithm with new genetic operators. The simulation results are shown in section 5 and finally, in section 6 we represent a conclusion and future works.

2 Voronoi Diagram

One of the most practical geometric structures is the Voronoi diagram space (*VD*). This geometric object is defined on the set of points in the *d*-dimensional ($d \geq 1$). *VD* includes all problems in nearest or farthest neighborhood problems [19].

Definition: Let $D = \{p_1, p_2, ..., p_n\}$ be a set of *n* points (or sites) in the plane. The Voronoi region of any point p_i, denoted by VR_i, is a convex polygon such that all points that lie in it, closer to p_i than other points of *D*. The locus of the points the plane that have more than one nearest site is called the Voronoi diagram of *D* and is denoted by *VD*(*D*).

Fortune's algorithm can construct the *VD* of *n* points in the plane in $O(n \log n)$ time complexity. The most important property of *VD* is that it can answer the nearest (or farthest) query point in $O(\log n)$ time. Also updating Voronoi diagram takes $O(n)$ time per point that has to be added or deleted [20-21].

3 Constrained Space *p*-Center and Coverage Problem

Let $D = \{p_1, p_2, ..., p_n\}$ be a set of n demand points in the plane and *ESR* be effective sense radius of sensors (or facilities). The goal is covering all demand points by minimum number of *ESR*-sensors; If *p* be this optimal number, then $C = \{c_1, c_2, ..., c_p\}$ can be a set of *p* facility points (or centers) that must be positioned among the points of *D*. One approach to solve the constrained coverage problem is that for all number of facilities, *p=1,2,…,n* or binary search in $O(\log n)$ times the *p*-center problem is solved and find the minimum *p* such that maximum radius of largest circle is equal to or less than *ESR*.

In the constrained version, there exist some locations that are usually modeled by polygons in which the centers cannot be positioned. Let s_1 be the fitness of the solution of *p*-center problem without considering the constraints and s_2 be the fitness of solution of the constrained p-center problem. It is clear that $s_2 \geq s_1$. It has been proved that minimum feasible covering circle can be determined by one of the two possible cases [14]:

- The feasible center is one of the closest points of the boundary line segments of the constraints to one of the demand points.
- The feasible center is one of the points of the intersection of the bisector of two demand points with the boundary of the constraints.

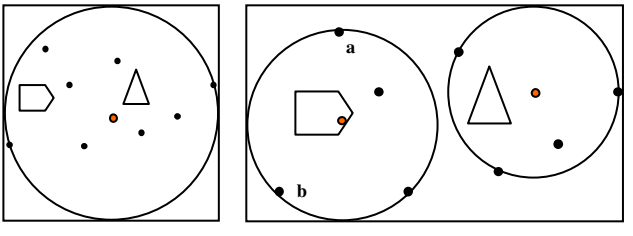


Fig. 1 Solving the *p*-center problem for two cases *p*=1 and 2

If *h* is the number of points on the convex hull and m is number of segments of constraints, finding the minimum feasible covering circle can be obtained in $O(m h^2)$ time. For example, fig. 1 shows two solutions of the unconstrained *p*-center problem for p=1 and 2. If we consider the constraints, the solution of case *p=1* is a feasible center ($s_2 = s_1$) but in *p=2* one of the two centers is infeasible

(center lies inside the pentagon constraint). In this case, it is sufficient to move the infeasible center along the bisector of the two points (a) and (b).

The *MCC* of n points in the plane can be computed in $O(n)$ computations using the Megiddo's parametric searching algorithm [25]. If this center is an infeasible (located in a constraints polygon) center can be changed to a feasible solution on the boundary of constraints. This changing runs in $O(m\,h^2)$ time, which m is the number of vertex in constraint polygons and h is the number of vertices in convex hull of n demand points. The convex hull of n points can be obtained in $O(n \log n)$ time [19]. In the *GA* we use above process, transferring each center to the boundary of constraint polygons, and satisfy constraint polygons in coverage problem.

In Addition to above constraints, in the constrained coverage problem there is another constraint; the *ESR* constraint. In fact the maximum sense radius of each sensor is limited to *ESR*, and this is a hard constraint. To satisfy the *ESR* constraint we use *GA* strategies and a tournament selection operator that emphasizes to the solutions that satisfy maximum sense radius.

4 Genetic Algorithm for Solving the Coverage Problem

*GA*s are probabilistic search and optimization techniques guided by the principles of evolution and natural genetics. This method provides global near-optimal solutions of an objective or fitness function by striking a remarkable balance between exploration and exploitation in complex, large, and multi-modal landscapes. There are several approaches to solve the constrained optimization problem using GAs [18], one of the popular methods is penalty function [23].

4.1 Parameters and Operators in Genetic Algorithm

4.1.1 Size of Population and Its Initialization

The size and initialization of population are two efficient factors in the success of evolutionary algorithms and reaching to the global solution. The size of population depends on the search space and the time can be spent. Usually, when there is no additional information about search space and its density, the population is initialized randomly. In the coverage problem if D is a set of demand points, the seed point solution can be selected in the convex hull of D. Since the optimal number of sensors is unknown, we first select a random number between one and n (number of demand points) as p, and then select p points in the convex hull of D randomly. Further more we can select randomly p points of n demand points (same as way that applied in discrete coverage problem). Also the initialization can be done by one of the seed point's p-center algorithms suggested in some studies [24]. It is notable that the size of chromosomes in the population is different and it is variant between one and n.

4.1.2 Selection Operator

The selection of parents to produce successive generations plays an extremely important role in the *GA*. The goal is to allow the fittest individuals to be selected more often to reproduce. However, all individuals in the population have a chance of being selected to reproduce the next generation. We use a new tournament selection operator by radius r. In this method we first select a random number as k, between 1 and *N*, which *N* is the size of population. Then we select the best solution between k^{th} and $(k+r)^{th}$ solution, called first parent. To select a pair for this parent, we repeat this process again. If the radius parameter r is a large number, the selection operator emphasizes the best solutions and increases the convergence rate, but this work may prevent the algorithm from reaching the global optimum solution. On the other hand, if r is a small, the solution of population spreads on the wide search space and the algorithm can find the global optimum solution by a high probability, but it is notable that the speed of progress of the population towards the better regions of search space is reduced. Anyway the parameter r can be used to balance exploration and exploitation. This selection operator is very simple to implement and proper in non-uniform and multi-modal search space, but it isn't promised for uniform and simple landscapes, opposite other selection operators likes Roulette wheel and proportional.

With regard to the goal of coverage problem; the fitness of each solution can be defined as the number of sensors that is used to cover. On the other hand the radius of sensors is limited to *ESR*, so if the maximum radius of each demand points from the nearest sensor is equal to or less than *ESR*, the solution is a feasible solution and otherwise it is an infeasible solution. The tournament selection operator first emphasizes to the feasible solutions and then to a solution that has the minimum sensor, If both are infeasible, a solution whose maximum radius is less than another solution is selected by the selection operator. For reaching this goal the selection operator use the domination concepts [18].

4.1.3 Crossover Operator

Here we choose each chromosome equal to a solution for the coverage problem, but since the number of sensors can be between one and n; so each solution of the population is a vector as $C = \{c_1, c_2, ..., c_p\}$, where $p \in \{1, 2, ..., n\}$ and $c_j = (x_j, y_j)$ is the location of j^{th} facility center in solution. If $C^1 = \{c_1^1, c_2^1, ..., c_p^1\}$ and $C^2 = \{c_1^2, c_2^2, ..., c_q^2\}$ are two parents selected by selection operator, the crossover of them makes as follows:

First for any member of C^1 such as c_i^1 we find the first and second nearest member in C^2, assume they are c_{i1}^2 and c_{i2}^2, respectively. Then we calculate the following equation:

$$d_i = \left(\left\| c_i^1 - c_{i2}^2 \right\| - \left\| c_i^1 - c_{i1}^2 \right\| \right), \text{ for } i = 1, 2, \ldots, p. \quad (1)$$

In the next step we sort the members of C^1 according to their above values in descending order. Then a member of C^1 with the maximum value of d_i and the

nearest member of C^2 are selected. Thereafter both are paired together and deleted from C^1 and C^2 respectively. This process, p times is repeated and all pairs are found. After this step a common crossover operator can be used and parents are combined and produce some offsprings [21]. Above process is repeated to C^2 again and q pairs are found. By using a dynamic data structure and dynamic *VD* each crossover operation can be run in $O\left(Max\left(p\log p\ ,q\log q\right)\right)$ time [23-24].

In the simulation of this study, after the pairing process was completed, we use a linear crossover operator and produce two children (offsprings) from each pair of parents. In fact, if c_i^1 and c_j^2 are two members that are paired in k^{th} iterations of the above process, $k=1,2,...,p$ (or $k=1,2,...,q$ for C^2), the linear crossover produces two children as follows:

$$child_k^1=\lambda c_i^1+(1-\lambda)c_j^2 \quad , \quad child_k^2=(1-\lambda)c_i^1+\lambda c_j^2 \tag{2}$$

Where λ is a random value between zero and one.

If C^1 and C^2 have p and q members respectively; the crossover operator produces two offsprings with p members and two offsprings with q members. We select the two best of them for the next population. If we select these two best solutions among the six solutions, we can implement the elitist strategy, too.

4.1.4 Mutation Operator

A mutation operator alters or mutates one parent by changing one or more variables in some way or by some random amount to form one offspring and is used to prevent undesirable convergence in population and also prevent the algorithm from getting stuck in a local optimum of the problem. We use a different strategy.

Assume $C=\{c_1,c_2,...,c_p\}$ is a child that is produced by crossover operator. Geometrically c_j is a circle by center (x_j,y_j) and minimum radius r_j such that covers all demand points in VR_j. Most probably only one of the p circles has a maximum radius. If this radius is denoted by $Z(C)$, then

$$Z(C)=\underset{1\le i\le n}{Max}\left\{\underset{1\le j\le p}{Min}\ d(p_i,c_j)\right\}$$

Indeed $Z(C)$ is the maximum effective radius of solution *C*. If the center and radius of the largest circle in *C* denoted by $c_{\max}$ and $r_{\max}$, respectively, then the mutation is defined such that:

If $r_{\max}>ESR$ we insert a sensor to *C*, otherwise we delete one of them. The insertion and deletion can be run randomly or regularly. For example, if we want to insert a sensor, we add it in the largest circle (in *ESR* distance of $c_{\max}$) and if we want to delete a sensor, we delete a center which has a minimum distance of nearest neighborhood centers. Like the selection operator, this mutation operator first emphasizes to feasible solutions and then to the minimum number of sensors.

4.2 Genetic Algorithm

The *GA* moves from generation to generation selecting and reproducing parents until a termination criterion is met. The most frequently used stopping criterion is a specified maximum number of generations. Another termination strategy involves population convergence criteria. In general GAs will force much of the entire population to converge to a single solution. When the sum of the deviations among individuals becomes smaller than some specified threshold the algorithm can be terminated.

In solving the coverage problem by *GA*, first some random solutions are generated, and by using Voronoi diagram of each solution and computing the *MCC* of each Voronoi region of centers, the fitness of each solution is computed. Then in each generation, selection, crossover and mutation operators as described above, are used and the next generation of population is created. Since it is possible that from two feasible parents some infeasible offsprings are produced, we first find MCC of all demand points that lie on the Voronoi region of each member of offspring and then move each member to the center of *MCC* (transfer operator). If this center lies on the infeasible location, we use the approach mentioned in section 2 and move it to the boundary of the constraint polygon such that minimum feasible circle is created.

The outlined of genetic algorithm is presented as follows:

Genetic Algorithms for Constrained Coverage Problem (*GACCP*)

Inputs: Demand set *D*; effective sense radius, *ESR*; and constraint polygons.
Output: A solution for the constrained continuous coverage problem.
Step1 Compute the convex hull of *D* (Demand points set) and initialize the first population in the convex hull of *D*;
Step2 For each solution of population as $C = \{c_1, c_2, ..., c_p\}$ run the following three sub steps:
Step2-1 Compute Voronoi diagram of *C*;
Step2-2 Find all demand points lying in the Voronoi region of c_j and put them in the S_j, for $j = 1, 2, ..., p$;
Step2-3 Compute the minimum feasible covering circle of S_j and set its center as the new c_j, for $j = 1, 2, ..., p$. Radius of circle j^{th} denoted by r_j and let $r_{\max} = \underset{1 \le j \le p}{Max}(r_j)$;
Step3 use selection, crossover and mutation operators, and generate the next generation of the population;
Step4 If the termination condition holds, return the best solution in the population and finish the algorithm; otherwise go to Step2.

4.2.1 Analysis of Algorithm

If the size of the population is *N*, the number of demand points is *n*, the number of vertices in constraints is m and *p* is the number of facility points, then the complexity of *GACCP* can be computed as follows:

Step1, initialization of population, runs in $O(pN)$ computations. In the worst case p equals n. For any solution of population, calculating the Voronoi diagram, finding the Voronoi region that each demand points belongs to and finding the minimum covering circle of points that lie in each Voronoi region can be run in $O(p \log p)$, $O(n \log p)$ and $O(n)$ time, respectively. If the center of *MCC* is an infeasible center, it can be changed to a feasible center in $O(m_i n_j^2)$ time, which m_i is the number of vertex of constraint polygon that the infeasible center lies in and n_j is number of demand points that lie in Voronoi region of the infeasible center.

The crossover and mutation operators run in $O(p \log p)$ computations. So the total time complexity of the constrained version of problem is $O(Nm n^2)$ and if we ignore the constraints, that will be equal to $O(Nn \log p)$ time.

Since *GACCP* uses a population of solutions, it is not very sensitive to a single solution. On the other hand *GACCP* profits from the two operator classes. Standard genetic operators and transfer operator that moves each sensor toward the center of *MCC* of each region. This mutual advantage of *GACCP* causes it to be very powerful in search and convergence goals in evolutionary algorithms. Another advantage is that *GACCP* can be used to solve the discrete version of coverage problem, too; it is sufficient that we redefine the crossover operator such that only use replacement strategy instead of linear crossover that used in equation (*2*).

5 Simulation Results

In this section, *GACCP* is used on several random test problems and the results are shown in Figure2. All test problems are dealt with in a rectangular shape of size 450×650 and have three constraint polygons.

The algorithm is run for various numbers of demand points, n, and effective sensor radius, *ESR*. In each run, first n random demand points are generated, and then *GACCP* is run with several predefined sensor radii. The termination condition is considered to have at most 100 iterations; *GACCP* is used with these parameters:

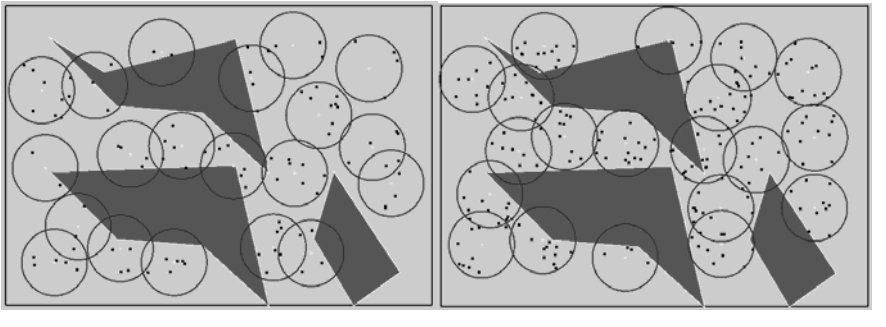


Fig. 2 Results of test problems for n=100 and 200 with *ESR*= 100

Size of population N=40, selection radius r=3, mutation probability is 1/p, which p is the number of sensors in solution, and we use local elitist strategy. Fig. 2 shows some results of the test problem for n=100 and 200 (Due to limitation in size of paper we don't represent more tests). Since, step2-3 of Algorithm changes all circles to minimum feasible circles, all solutions in end of each generation are feasible. The simulations need a little time to run between 20 to 30 seconds using *C#* programming on a home PC.

6 Conclusion

The presented new method combined the genetic algorithm with an elegant geometric approach to constrained coverage problem. Presented genetic operators are new which are suggested regarding the coverage problem goal. In addition, the selection operator that is used here is a new type of tournament selection operators. This operator can be used in any genetic algorithm.

The proposed algorithm can be used for solving the discrete and weighted coverage problem. With respect to the improvements being made to the weighted Voronoi diagram, naturally one of the future works can be presenting a compatible version of these algorithms in the non-uniform density space.

References

1. Yang, S., Dai, F., Cardei, M., Wu, J.: On connected multiple point coverage in wireless sensor networks. Journal of Wireless Information Networks 13(4), 289–301 (2006)
2. Bagheri, M.: Efficient k-coverage Algorithms for wireless sensor networks and their applications to early detection of forest. Master thesis, Simon Fraser University (2007)
3. So, A.M.-C., Ye, Y.: On solving coverage problems in a wireless sensor network using voronoi diagrams. In: Deng, X., Ye, Y. (eds.) WINE 2005. LNCS, vol. 3828, pp. 584–593. Springer, Heidelberg (2005)
4. Ye, F., Zhong, G., Lu, S., Zhang, L.: PEAS: A robust energy conserving protocol for long-lived sensor networks. In: Int'l Conf. on Distributed Computing Systems (ICDCS) (2003)
5. Murray, A.T., O'Kelly, M.E., Church, R.L.: Regional service coverage modeling. Computer and Operation Research 35, 339–355 (2008)
6. Hochbaum, D.S., Maass, W.: Fast approximation algorithms for a nonconvex covering problem. J. Algorithms 8, 305–323 (1987)
7. Current, J., O'Kelly, M.: Locating emergency warning sirens. Decision Sciences 23, 221–234 (1992)
8. Murray, A.T., O'Kelly, M.E.: Assessing representation error in point-based coverage modeling. Journal of Geographical Systems 4, 171–191 (2002)
9. Mladenovi´c, N., Labbe, M., Hansen, P.: Solving the p-center problem with tabu search and variable neighborhood search. Networks 42, 48–64 (2003)
10. Elloumi, S., Labbe, M., Pochet, M.: A new formulation and resolution method for the p-center problem. INFORMS Journal on Computing 16(1), 84–94 (2004)

11. Chen, D., Chen, R.: New relaxation-based algorithms for the optimal solution of the continuous and discrete p-center problems. Computers and Operations Research (2008), doi:0.1016/j.cor.2008.03.009
12. Daskin, M.S.: Network and Discrete Location: Models, Algorithms and Applications. John Wiley, New York (1995)
13. Drezner, Z.: The p-center problem- heuristics and optimal algorithms. Journal of the Operational Research Society 35, 741–748 (1984)
14. Wei, H., Murray, A.T.: Solving the continuous space p-centre problem: planning application issues. IMA Journal of Management Mathematics 17, 413–425 (2006)
15. Huang, C., Tseng, Y.: The Coverage Problem in a Wireless Sensor Network. Mobile Networks and Applications 10, 519–528 (2005)
16. Hall, D.L., Llinas, J.: Handbook of Multisensor Data Fusion. CRC Press, Boca Raton (2001)
17. Zhao, Z., Govindan, R.: Understanding packet delivery performance in dense wireless sensor networks. In: Proc. of The Third ACM Conference on Embedded Networked Sensor Systems (Sensys 2003), Los Angeles, CA, pp. 1–13 (2003)
18. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley, Chichester (2001)
19. O'Rourke, J.: Computational Geometry in C, 2nd edn. Cambridge University Press, New York (1998)
20. Gowda, I., Kirkpatrick, D., Lee, D., Naamed, A.: Dynamic Voronoi Diagrams. IEEE Transactions on Information Theory 29, 724–731 (1983)
21. Borglet, M.G., Borglet, C.: Notes on the Dynamic Bichromatic All-Nearest-Neighbors Problem. In: 23rd European Workshop on Computational Geometry, pp. 198–201 (2007)
22. Megiddo, N.: Linear-time algorithms for linear programming in R3 and related problems. SIAM Journal on Computation 12, 759–776 (1983)
23. Homaifar, A., Lai, S.H., Qi, X.: Constrained optimization via genetic algorithms. Simulation 62(4), 242–254 (1994)
24. Pelegrin, B., Canovas, L.: A new assignment rule to improve seed points algorithms for the continuous k-center problem. European Journal of Operational Research 104, 366–374 (1998)

Using Multi-objective Evolutionary Algorithms in the Optimization of Polymer Injection Molding

Célio Fernandes, António J. Pontes, Júlio C. Viana, and A. Gaspar-Cunha

Abstract. A Multi-objective optimization methodology has been applied in the optimization of polymer injection molding process. This allowed the optimization of the operating conditions of the process from mold flow simulations, taking into consideration the existence of 5 criteria simultaneously, such as temperature difference on the molding at the end of filling, the maximum pressure, the pressure work, the volumetric shrinkage and the cycle time. The results produced shown that the proposed methodology is an efficient tool to be used in the optimization of this process.

1 Introduction

The injection molding process is one of the most important polymer processing technologies used to manufacture a great variety of plastics parts of high complexity and with tight dimensional tolerances. Injection molding of polymeric materials is a complex process involving several phases, such as plasticating (solids conveying, melting), melt flow (injection), pressurization (holding) and cooling/solidification. This, together with the viscoelastic nature of the polymeric materials, strongly affects the quality of the final molded parts.

The thermomechanical environment imposed to the polymer is determined by the operating conditions, the system geometry and the polymer properties. The final part morphology obtained, which determines their dimensions, dimensional stability and final properties, is controlled by the thermomechanical conditions defined by the process [1, 2]. Therefore, due to the high number of variables involved and the strong interaction between them and the end product properties, the definition of the best operating conditions to use in a specific processing situation is a complex task. Sophisticated modeling programs, able to predict the process response to the operating conditions defined, have been used for that purpose in an iterative way [3, 4]. The user defines some tentative processing conditions for the process under study and then these solutions are evaluated using the modeling routines in terms of the relevant criteria previously defined. If the desired response is not satisfactory the user defines new operating conditions, taking into account the

Célio Fernandes · António J. Pontes · Júlio C. Viana · A. Gaspar-Cunha
IPC- Institute for Polymer and Composites, Dept. of Polymer Engineering,
University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal
e-mail: {cbpf, pontes, jcv, agc }@dep.uminho.pt

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 357–365.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

results obtained for the previous ones, and the process continues until an acceptable performance is attained [3, 4].

However, this trial-and-error process is strongly dependent on the ability of the user to define the operating conditions to be tested. Thus, it is of great importance the application of an automatic optimization methodology able to define the operative window for the injection molding process without minimal user intervention.

Various optimization strategies using different methodologies to optimize the injection molding process have been reported in the literature [5-9]. Kim *et al.* [5] used Genetic Algorithms (GA) to optimize the processing variables (mold and melt temperatures and filling time) based on pre-defined criteria. The performance of the process was quantified using a weighted sum of the temperature difference, "overpack" and frictionally overheating criteria. Lotti and Bretas [6] applied Artificial Neural Networks, ANN, to predict the morphology and the mechanical properties of an injection molded part as a function of the processing conditions (mould and melt temperatures and flow rate). A central composite design of experiments approach was used to predict the molding morphology as a function of the processing conditions using the MoldFlow software. Castro *et al.* [7] used ANN and data envelopment analysis (*i.e.*, statistical analysis) to find the optimal compromise between multiple performance measures to define the setting of injection molding variables and the gate location. Peic and Turng [8] used three different optimization algorithms (evolution strategies, differential evolution and simulated annealing) to optimize the injection molding processing conditions as a function of cycle time and volumetric shrinkage, using as restrictions the clamping force, the injection pressure and the temperature of the part at ejection. The relation between the processing conditions and the optimization criteria was performed with the CMOLD software. Finally, Alam *et al.* [9] applied a Multi-Objective Evolutionary Algorithm (MOEA) to optimize the shrinkage of the molding and perform the runner balancing. Gaspar-Cunha and Viana [10] optimized the mechanical properties of injection molded parts using an MOEA approach.

In this work an optimization methodology based on MOEA is used for optimizing the operating conditions (melt and mold temperatures, injection flow rate, switchover point, holding time and pressure) in injection molding. This problem was not solved in the previous works above identified. A study about the performance of the proposed methodology using various optimization criteria has been carried out. The MOEA was linked with an injection molding simulator (CMOLD), which is able to compute the optimization criteria as a function of the defined operating conditions and able to take into account, simultaneously, the system geometry and polymer properties.

2 Multi-objective Evolutionary Algorithms

The use of computer simulations on the design stages of engineering plastic components for the injection molding process is very frequent [3, 4]. Initially, a finite element mesh representative of the part geometry is defined, the materials are selected, the gate location is defined and the initial processing variables are introduced. Then, after launching the simulation the outputs are analyzed. A trial-and-error process is applied, where the initial conditions, in what concerns geometry,

material and/or processing conditions are modified until the desired results are obtained. This process can be very complex since in most cases multiple criteria are to be optimized simultaneously. Also, the finding of a global optimum solution is not guaranteed.

The methodology proposed in this work integrates the computer simulations of the injection molding process, an optimization methodology based on evolutionary algorithm and multi-objective criteria in order to establish the set of operative processing variables leading to a good molding process. The optimization methodology adopted is based on a Multi-Objective Evolutionary Algorithm (MOEA) [11], due to multi-objective nature of most real optimization problems, where the simultaneous optimization of various, often conflicting, criteria is to be accomplished [10-15]. The solution must then result from a compromise between the different criteria. Generally, this characteristic is taking into account using an approach based on the concept of Pareto frontiers (*i.e.*, the set of points representing the trade-off between the criteria) together with an MOEA. This enabled the simultaneously accomplishment of the several solutions along the Pareto frontier, *i.e.*, the set of non-dominated solutions [14, 15].

The link between the MOEA and the problem to solve is made in two different steps (see flowchart of Fig. 1). First, the population is randomly initialized, where each individual (or chromosome) is represented by the binary value of the set of all variables. Then, each individual is evaluated by calculating the values of the relevant criteria using the modeling routine (in this case CMOLD). Finally, the remaining steps of a MOEA are to be accomplished. To each individual is assigned a single value identifying its performance on the process (fitness). This fitness is calculated using a Multi-Objective approach as described in details elsewhere [12, 15]. If the convergence criterion is not satisfied (*e.g.*, a pre-defined number of generations), the population is subjected to the operators of reproduction (*i.e.*, the selection of the best individuals for crossover and/or mutation) and of crossover and mutation (*i.e.*, the methods to obtain new individuals for the next generation). The RPSGAe uses a real representation of the variables, a simulated binary crossover, a polynomial mutation and a roulette wheel selection strategy [11, 14].

3 Injection Molding Optimization

The optimization methodology proposed will be used for setting the processing conditions of the molding shown in Fig. 2, which will be molded in polystyrene (STYRON 678E). The relevant polymer properties used for the flow simulations were obtained from the software (CMOLD) database. A mesh with 408 triangular elements has been used. The simulations considered the mold filling and holding (post-filling) stages. A node near the P1 position, pressure sensor position (see Fig. 2) was selected as a reference point for this study. The processing variables to optimize and allowed to varied in the simulations in following intervals were: injection time, $t_{inj} \in [0.5; 3]$ s (corresponding to flow rates of 24 and 4 cm^3/s, respectively), melt temperature, $T_{inj} \in [180; 280]$ ºC, mold temperature, $T_w \in [30; 70]$ ºC, holding pressure, $Ph \in [7; 38]$ % of maximum machine injection pressure,

with fixed switch-over point, SF/P, at 99%, holding time, t2P, of 30s and timer after hold pressure of 15s.

For the results produced by the modeling programme two process restrictions were imposed: i) the molding has to be completely filled, obviously no short-shots

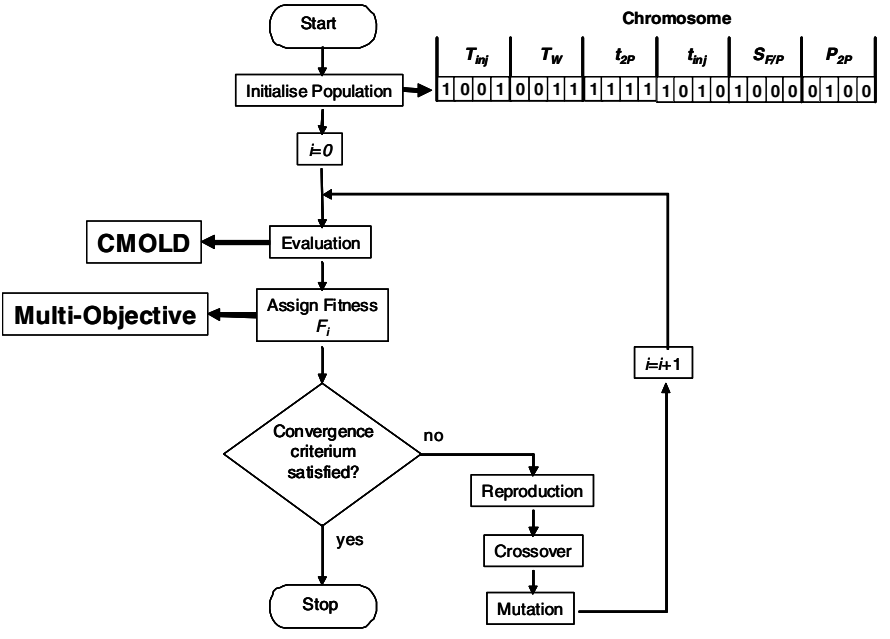


Fig. 1 Flowchart of the MOEA applied for the optimization of the injection molding process

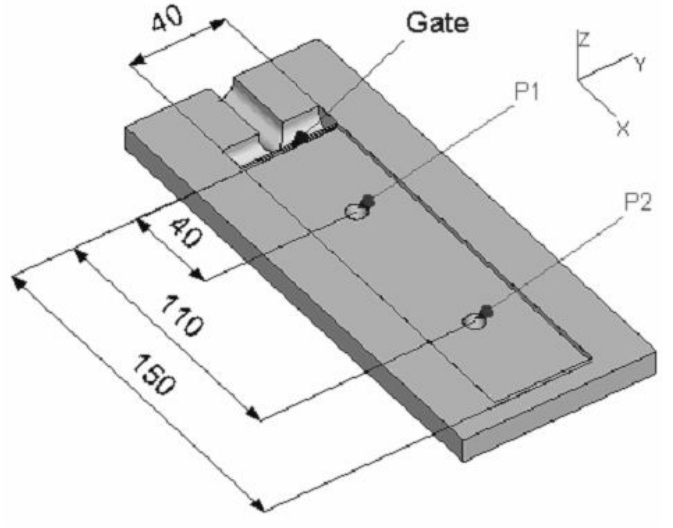


Fig. 2 Die insert of the injection molding part with 2 mm of thickness (dimensions in mm)

were admitted and ii) the computed values of the maximum shear stress and strain-rate were limited to their critical values (defined on the CMOLD database) in order to avoid potential defects (*e.g.*, melt fracture).

The criteria used are the following:

1) The temperature difference on the molding at the end of filling, $dT = (T_{max}\text{-}T_{min})$, was minimized to avoid part distortions and warpage due to different local cooling rates, $dT \in [0; 20]$ ºC;
2) The volumetric shrinkage of the moldings was minimized, $VS \in [0; 15]$ %;
3) The maximum mold pressure was minimized, reducing the clamping force, $P_{max} \in [1; 70]$ MPa;
4) The cycle time was minimized, increasing productivity, $t_c \in [30; 35]$ s;
5) The pressure work, defined as the integral of pressure, P, along the time, t:

$$PW = \int_0^{t_c} P(t)dt \tag{1}$$

was to be minimized in order to diminish the residual stress, the energy consumption and to reduce the mechanical efforts supported by the equipment, $PW \in [0; 200]$ MPa.s.

The RPSGAe was applied using the following parameters: 50 generations, cross-over rate of 0.8, mutation rate of 0.05, internal and external populations with 100 individuals, limits of the clustering algorithm set at 0.2 and NRanks = 30. These values resulted from a carefully analysis made in a previous paper [11]. Since the aim of this work is to study the applicability of the optimization methodology in the injection molding process taking into account the various criteria identified above, the 20 optimization runs identified in Table 1 were carried out. Runs 1 to 9 considered 2 criteria simultaneously, runs 10 to 19 considered 3 criteria and run 20 all the 5 criteria.

Table 1 Criteria used in each process optimization run

Run	Optimized criteria	Run	Optimized criteria
1	*VS* and *dT*	11	*VS*, *dT* and t_c
2	*VS* and P_{max}	12	*VS*, *dT* and *PW*
3	*VS* and t_c	13	*VS*, P_{max} and t_c
4	*VS* and *PW*	14	*VS*, P_{max} and *PW*
5	*PW* and *dT*	15	*VS*, t_c and *PW*
6	*PW* and P_{max}	16	*PW*, *dT* and P_{max}
7	*PW* and t_c	17	*PW*, *dT* and t_c
8	t_c and *dT*	18	*PW*, P_{max} and t_c
9	t_c and P_{max}	19	t_c , *dT* and P_{max}
10	*VS*, *dT* and P_{max}	20	All criteria

4 Results and Discussion

Fig. 3 shows the results obtained when 2 criteria are used simultaneously (for runs 4 to 7 as an example). The optimization algorithm is able to evolve during the 50 generations and to produce a good approximation to the Pareto frontier in all the cases.

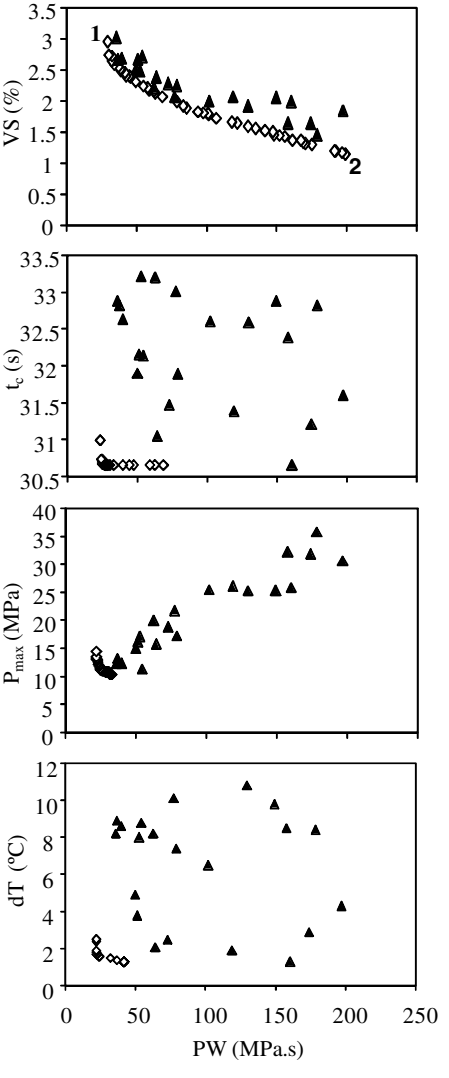


Fig. 3 Optimization results for two criteria, runs 4 to 7. Open symbols: Pareto frontier at 50th generation; full symbols: initial population (*PW* – Power Work, *VS* – volumetric shrinkage, t_c – cycle time, P_{max} – maximum pressure, *dT* – temperature difference)

For example, if run 4 is considered (*VS* – *PW* plot of Fig. 3), when solution 1 (*VS* = 3% and *PW* = 29.8 MPa.s) is compared with solution 2 (*VS* = 1.1% and *PW* = 199 MPa.s) in what concerns the operating conditions obtained, the most relevant changes are the injection temperature and the holding pressure (solution 1: t_{inj} = 2.6 s, T_{inj} = 207 ºC, T_w = 31 ºC and *Ph* = 7%; solution 2: t_{inj} = 2.6 s, T_{inj} = 223 ºC, T_w = 30 ºC and *Ph* = 29 %). Both changes contribute to the increase of *PW* and consequently to the decrease of *VS*. The increment of T_{inj} decreases the melt viscosity, promoting a better packing of the material during the holding phase, so reducing shrinkage. Furthermore, increasing *PW* does not increase the cycle time as would be expected. Interestingly, P_{max} and *dT* both slightly decrease with *PW*, but the changes on these variables is small.

Fig. 4 shows that when 3 criteria are used (Fig. 4 – left), a compromise between all the criteria considered is attained. Some of the Pareto solutions produced seem to be dominated in the individual 2 criteria plots. Identical conclusion can be made when 5 criteria are used (Fig. 5 – left): the Pareto frontier results from the compromise between all the 5 criteria. Figures 4 and 5 show also that the optimization algorithm is able to attain good solutions when compared with the 2 criteria cases (Figs. 4 and 5 – right).

Finally, the selection of a single solution (from the set of non-dominated solutions) to use in the real process depends on the articulation of preferences between the different criteria [16]. Due to its complexity, this is not subject of this text but will be considered in a near future [16].

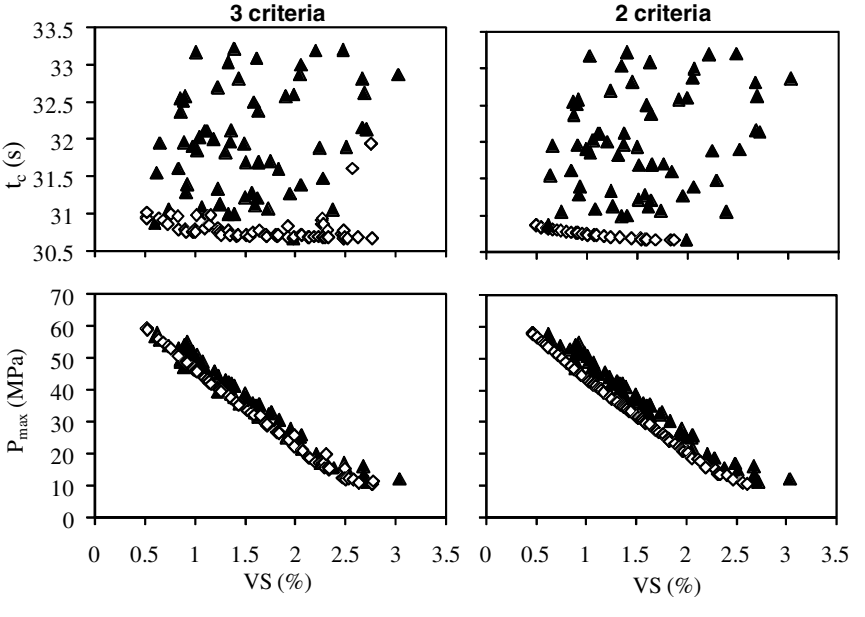


Fig. 4 Optimization results for 3 criteria of run 13 – left; and comparison with runs 2 and 3 – right (open symbols: Pareto frontier at 50th generation; full symbols: initial population)

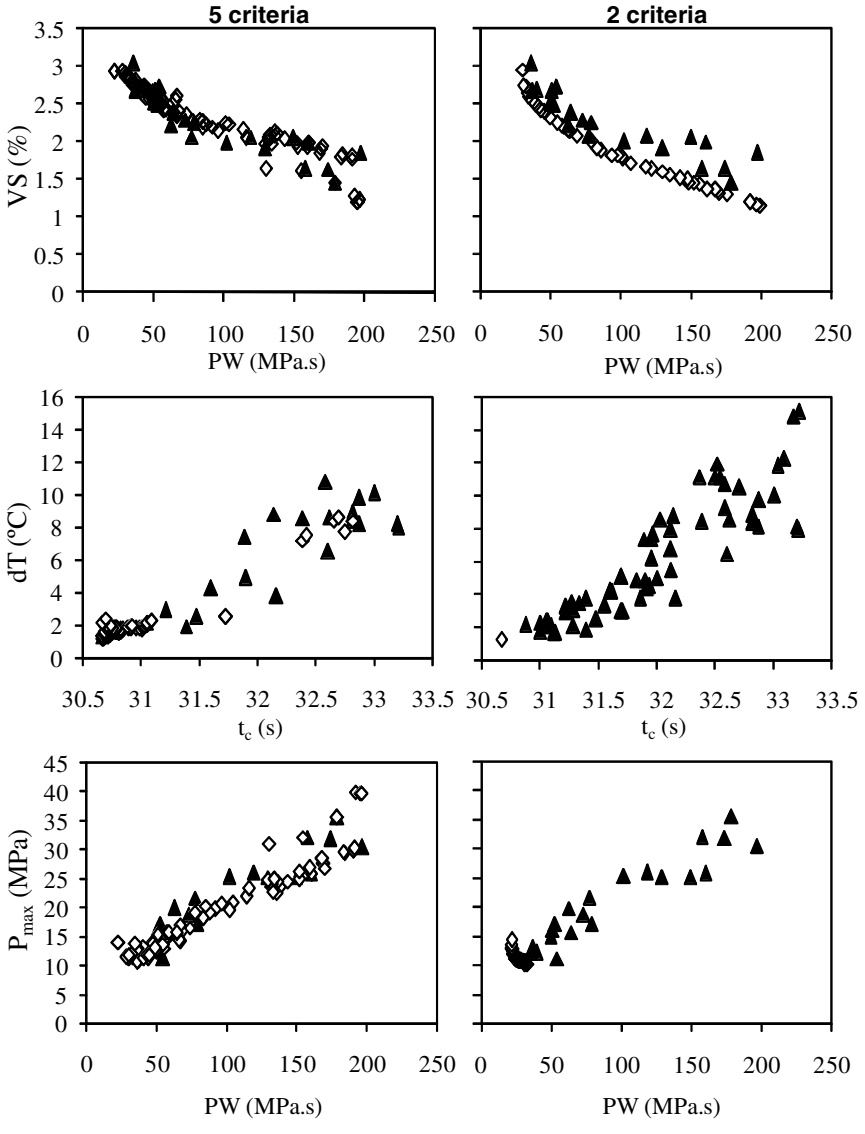


Fig. 5 Optimization results for 5 criteria of run 20 – left; and comparison with runs 4, 6 and 8 – right (open symbols: Pareto frontier at 50th generation; full symbols: initial population)

5 Conclusions

A multi-objective optimization methodology based on Evolutionary Algorithms (MOEA) was applied in the optimization of the processing conditions of polymer injection molding process. The algorithm is able to take into account the multiple

criteria used and, with a single run, to obtain a complete trade-off of solutions. The results produced have physical meaning which can be justified by the analysis of the process.

Acknowledgments. This work was supported by the Portuguese Fundação para a Ciência e Tecnologia under grant SFRH/BD/28479/2006.

References

1. Viana, J.C., Cunha, A.M., Billon, N.: The thermomechanical environment and the microstructure of an injection moulded polypropylene copolymer. Polym. 43, 4185–4196 (2002)
2. Viana, J.C.: Development of the skin layer in injection moulding: phenomenological model. Polym. 45, 993–1005 (2004)
3. Cmold User´s manual
4. Moldflow User´s manual
5. Kim, S.J., Lee, K., Kim, Y.I.: Optimization of injection-molding conditions using genetic algorithm. In: Proc of SPIE – the 4th Int Conf on Computer-Aided Design and Computer Graphics, vol. 2644, pp. 173–180 (1996)
6. Lotti, C., Bretas, R.E.S.: The influence of morphological features on mechanical properties of injection molded PPS and sits prediction using neural networks. In: Proc. of PPS, p. 184 (2003)
7. Castro, C., Bhagavatula, N., Cabrera-Rios, M., et al.: Identifying the best compromises between multiple performance measures in injection molding (IM) using data envelopment analysis. In: Proc. SPE technical papers ANTEC 2003 (2003)
8. Turng, L.S., Peic, M.: Computer aided process and design optimization for injection molding. J. of Eng. Manuf. Proc. of the Inst. of Mech. Eng (Part B) 12, 1523–1532 (2002)
9. Alam, K., Kamal, M.R.: A genetic optimization of shrinkage by runner balancing. In: Proc. SPE technical papers ANTEC 2003, pp. 637–641 (2003)
10. Cunha-Gaspar, A., Viana, J.C.: Using multi-objective evolutionary algorithms to optimize mechanical properties of injection moulded parts. Int. Polym. Proc. XX 3, 274–285 (2005)
11. Gaspar-Cunha, A., Covas, J.A.: RPSGAe – reduced pareto set genetic algorithm: application to polymer extrusion. In: Gandibleux, X., Sevaux, M., Sörensen, K., et al. (eds.) Metaheuristics for multiobjective optimization. Springer, Heidelberg (2004)
12. Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Boston (1989)
13. Gaspar-Cunha, A.: Modelling and optimization of single screw extrusion. Ph D Thesis, University of Minho, Braga (2000)
14. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, New York (2001)
15. Coello Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary algorithms for solving multi-objective problems. Kluwer, Massachusetts (2002)
16. Ferreira, J.C., Fonseca, C.M., Gaspar-Cunha, A.: Methodology to select solutions from the pareto-optimal set: a comparative study. In: GECCO 2007 Genetic and Evolutionary Computation Conference, pp. 789–796 (2007)

A Multiobjective Extremal Optimization Algorithm for Efficient Mapping in Grids

Ivanoe De Falco, Antonio Della Cioppa, Domenico Maisto, Umberto Scafuri, and Ernesto Tarantino

Abstract. Extremal Optimization is proposed to map the tasks making up a user application in grid environments. To comply at the same time with minimal use of grid resources and maximal hardware reliability, a multiobjective version based on the concept of Pareto dominance is developed. The proposed mapper is tested on eight different experiments representing a suitable set of typical real–time situations.

1 Introduction

Grid is a decentralized heterogeneous multisite system which, aggregating multi–owner resources spread across multiple domains, creates a single powerful collaborative problem–solving environment. Thus, given a grid constituted by several sites, each containing one or more nodes, the communicating tasks of a parallel application could be conveniently assigned to the grid nodes which, selected on the basis of their features and load conditions, result to be the most suitable to execute them. However, when parallel applications are executed together with the local workloads, i.e. on non–dedicated grid systems, maximizing parallel efficiency and minimizing influence on performance is an open challenge. Obviously, in absence of information about the communication timing, the co–scheduling of the communicating tasks of a parallel application must be guaranteed to avoid possible deadlock conditions [1].

Often the search for one single site onto which to map all the tasks making up the application could fail to fulfil all these needs. Thus, a multisite mapping tool, able

Ivanoe De Falco · Domenico Maisto · Umberto Scafuri · Ernesto Tarantino
ICAR - CNR, Via P. Castellino 111, 80131 Naples, Italy
e-mail: last_name.initial_of_first_name@na.icar.cnr.it

Antonio Della Cioppa
Natural Computation Lab, DIIIE, University of Salerno, Via Ponte don Melillo 1, 84084 Fisciano (SA), Italy
e-mail: adellacioppa@unisa.it

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 367–377.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

to choose among resources spread over multiple sites and to match applications demands with the grid computing resources, must be designed.

Our mapper is designed to perform a multiobjective optimization and distribute the application tasks among the nodes minimizing the use of grid resources and, at the same time, satisfying also Quality of Service (QoS) [2] requirements, in particular maximizing reliability by preferring devices, i.e. processors and links connecting the sites to internet, which only seldom are broken. Since mapping is an NP–complete problem [3], several evolutionary–based techniques have been used to face it in a heterogeneous or grid environment [4, 5, 6]. Although the problem shows correlations among variables due to communications, the use of a parameter–wise evaluation of the objective function is possible. Therefore we introduce in literature Extremal Optimization (EO) [7, 8], a coevolutionary algorithm, to face mapping.

Besides, to provide the user with a set of mapping solutions, each with different balance for use of resources and reliability, a multiobjective version of EO based on the Pareto method [9] is here designed. Peculiar issues of our mapper are a multisite approach and the view of the nodes making up the sites as the lowest computational units taking their reliabilities and loads into account.

In the following, section 2 outlines EO method and its multiobjective version developed by us and section 3 describes our view on the mapping problem and its formalization in terms of EO. Section 4 reports on the test problems experienced and shows the findings achieved, while section 5 contains our conclusions.

2 Extremal Optimization

The Bak–Sneppen model [10] simulates an ecosystem based on the principle that evolution progresses by selecting against the few most poorly adapted species, rather than by breeding those species well adapted to the environment. According to this model, each component of the ecosystem corresponds to a species which is characterized by a fitness value. The evolution is driven by a process where the least fit species is selected for a random update and also its closest dependent (*i.e.* most correlated) species have their fitness replaced. EO draws upon the Bak–Sneppen mechanism and represents a successful method for the study of NP–hard problems [7, 8]. For minimization problems, our modified version of EO proceeds as in Algorithm 1. It works with a single solution S made of a given number of components s^i, each of which is a species representing a variable of the problem and is assigned a fitness value ϕ^i. At each time step, S is evolved by randomly updating its worst component. We choose the "worst" component probabilistically. During our preliminary experiments for multiobjective mapping the selection probability used in the probabilistic version of EO (*i.e.* τ-EO) based on a parameter τ depending on problem size has led to behaviours ranging from random walk dynamics to deterministic ones for very small variations of τ. Therefore we have preferred to use an exponential ranking, which has shown more stable dynamics, so we order the species for increasing fitness values, then we assign to the generic species with i–th position in ranking a probability of being selected $p_i = q \cdot (1-q)^i$ where q is

Algorithm 1. Pseudocode of the EO algorithm

```
begin
  initialize solution S at will
  set S_best := S
  while maximum number of iterations N_iter not carried out do
    evaluate φ^i for each variable s^i of the current solution S
    rank the variables s^i based on their fitness φ^i
    choose probabilistically the variable s^j with the worst fitness
    choose S' ∈ Neighbour(S) such that s^j must change
    accept S := S' unconditionally
    if Φ(S) < Φ(S_best) then
      set S_best := S
    end if
  end while
  return S_best and Φ(S_best)
end
```

a real number in $[0.0, 1.0]$ and a parameter of EO. Once chosen probabilistically the worst species in S, we have to perform a "move" to get to a neighboring solution $S' \in Neighbour(S)$. Since moving a task has a direct influence on the communications it has to perform with the other tasks, we define a neighbor S' of a given solution S as one obtained by moving a task T_i in S, chosen based on the above mechanism, from a site $\mathscr{S}_i$ to a random node in another site $\mathscr{S}_k$, and by moving to random nodes belonging to $\mathscr{S}_k$ also any other task which communicates with T_i more than it does with any other task. Then S' becomes the new solution S, its overall fitness Φ is computed and, if is better than the current best S_{best}, replaces it. These actions are repeated for a maximum number of iterations $\mathscr{N}_{iter}$.

2.1 Multiobjective Extremal Optimization

Since mapping is a multiobjective problem, it can be profitably faced by using an approach based on the concept of the so–called *Pareto optimal set* [9]. With this mechanism, a set of "optimal" solutions emerges in which none of them can be considered to be better than any other in the same set.

The EO mechanism is here modified to face multiobjective problems. Let us suppose to have two objectives to optimize. At each time step, the current solution variables are ranked for any objective as in Algorithm 1. The aim is to choose either one or two of the different rankings and, consequently, get the related "worst" variable(s) changed. So we can push the search mechanism towards either one or two objectives. This has been done because preliminary experiments have shown that executions attempting at optimizing at the same time both the objectives at any iteration lead to evolutions trapped in the central part of the theoretical Pareto front, and its extremal points, *i.e.* the best solutions for the two objectives, are never reached. This "saw–tooth" behaviour, instead,

allows to obtain the whole Pareto front and the best solutions as well. The two objectives are mapped in the interval $[-1.0, 1.0]$ to three contiguous segments $\mathscr{I}_1 = [-1, -0.5[$, $\mathscr{I}_2 = [-0.5, 0.5]$ and $\mathscr{I}_3 =]0.5, 1.0]$. The first objective is associated to $\mathscr{I}_1$, the second one to $\mathscr{I}_3$ and, finally, both the objectives are linked to $\mathscr{I}_2$. The above values are the result of preliminary experiments. Moreover, a real number r is computed as $r = (1 - \frac{t}{\mathscr{N}} * 2)$, where $\mathscr{N}$ is a parameter for EO and t is the current time step within $[0, \mathscr{N} - 1]$. The objective(s) associated to the segment into which r falls is(are) selected and at each EO iteration the related "worst" variable(s) is(are) changed. For any iteration a new solution is proposed which can enter the current Pareto set and replace the solutions it dominates in that set. As the number of iterations increases better and better solutions will likely be found and stored according to the Pareto criterion so that the Pareto front will shift towards the theoretically best one.

3 Mapping in Grids

We assume to have an application subdivided into P tasks to be mapped on n ($n \leq P$) nodes. Each node is identified by an integer value in the range $[1, N]$, where N is the total number of available grid nodes.

The number of instructions α_i computed per time unit on each node i, and the communication bandwidth β_{ij} between any couple of nodes i and j must be known *a priori*. This information is supposed to be either statistically estimated in a particular time span or collected by tracking periodically and by forecasting dynamically resource conditions [12, 13]. For example, in Globus [2], a standard grid middleware, similar information is gathered by the Grid Index Information Service (GIIS) [13].

As grids address non–dedicated resources, their local workloads must be considered to evaluate the computation time of the tasks. Prediction methods exist to take account of non–dedicated resources [14, 15]. We suppose to know the average load $\ell_i(\Delta t)$ of the node i at a given time span Δt with $\ell_i(\Delta t) \in [0.0, 1.0]$, where a higher value means a higher load. Hence $(1 - \ell_i(\Delta t)) \cdot \alpha_i$ is the power fraction of the node i available for execution of grid tasks.

As regards the resources requested by the application task, we assume to know for each task k the number of instructions γ_k and the amount of communications ψ_{km} between the k–th and the m–th task $\forall m \neq k$. This information can be obtained either by a static program analysis or by using smart compilers or by other tools able to generate them. For instance the Globus Toolkit includes an XML standard format to define application requirements [13].

Finally, information must be known about the degree of reliability of any component of the grid in terms of fraction of actual operativity π_z for the processor z and λ_w for the link connecting to internet the site w to which z belongs. These values can be gathered by means of a historical and statistical analysis, and range in $[0.0, 1.0]$ where a higher value means a better reliability.

Encoding. Any mapping solution is represented by a vector μ of P integers ranging in the interval $[0, N-1]$, whose generic component $\mu_i = j$ means that the solution maps the i–th task of the application onto node j of the grid.

Fitness. We have the following two fitness functions, one for each objective.

Use of Resources. Denoting with τ_{ij}^{comp} and τ_{ij}^{comm} respectively the computation and the communication times to execute the task i on the node j it is assigned to, the total time needed to execute i on j is $\tau_{ij} = \tau_{ij}^{\text{comp}} + \tau_{ij}^{\text{comm}}$. It is evaluated on the basis of the computation power and of the bandwidth which remain available once considered the local and grid workloads. We set $\phi_1^i = \tau_{ij}$ for each task. The fitness function for the use of resources is then the maximum among those values:

$$\Phi_1(\mu) = \max_{i \in [1,P]} \{\phi_1^i\} \tag{1}$$

This definition of the above parameter–wise objective function aims to search for the smallest fitness value among these maxima, i.e. to find the mapping which uses at the minimum, in terms of time, the grid resource it has to exploit at the most.

Reliability. If μ_i denotes the node onto which the i–th task is mapped and w the site this node belongs to, the fitness function ϕ_2^i for the i–th component is equal to $\pi_{\mu_i} \cdot \lambda_w$. Then, the fitness function for the whole solution is:

$$\Phi_2(\mu) = \prod_{i=1}^{P} \pi_{\mu_i} \cdot \lambda_w \tag{2}$$

The first fitness function should be minimized and the second maximized. We face this two–objective problem with a Multiobjective EO (MEO) algorithm based on the Pareto–front approach, as described in section 2.1; at the end of the run we propose to the user all the solutions present in the final *Pareto set*.

4 Experimental Results

For our experimental framework we assume a multisite grid composed of ten sites with a total of $N = 184$ nodes, indicated by the external numbers from 0 to 183 in Fig. 1. For example 50 in Fig. 1 is the third of the 8 nodes in the site C. The values for the computing powers, the communication bandwidths and the load conditions of the grid nodes, and the hardware reliabilities should be chosen conveniently since it is in this way easier, by arranging suitably experiments, to know the optimal solutions and thus assess the goodness of the solutions achieved by MEO.

Without loss of generality we suppose that all the nodes in a same site have the same power α. For example all the nodes belonging to C have $\alpha = 1500$ MIPS.

As for the bandwidth, we denote with β_{ii} the one for two communicating tasks being executed on the same node i, and with β_{ij} that for two tasks being executed one on the node i and the other on the node j. The former represents the *intranode* communication, whereas the latter denotes either the *intrasite* (when the nodes i and j belong to the same site) or the *intersite* communication (when the nodes i

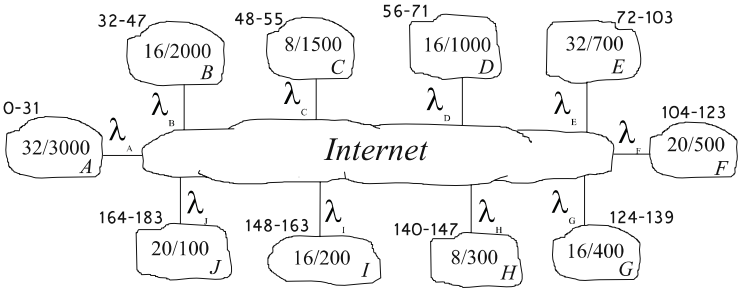


Fig. 1 The grid architecture

Table 1 Intersite and intrasite bandwidths expressed in Mbit/s

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
A	100									
B	2	200								
C	2	2	200							
D	4	4	4	400						
E	4	4	4	4	400					
F	8	8	8	8	8	800				
G	16	16	16	16	16	16	1000			
H	16	16	16	16	16	16	16	1000		
I	32	32	32	32	32	32	32	32	2000	
J	32	32	32	32	32	32	32	32	32	2000

Table 2 Reliability for the nodes

Sites	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
π	0.90	0.92	0.94	0.96	0.98	0.99	0.97	0.95	0.93	0.91

and j belong to different sites)(Table 1). For the sake of simplicity we presume that $\beta_{ij} = \beta_{ji}$ and that $\beta_{ii} = 100$ Gbit/s $\forall i \in [1, N]$.

Moreover we suppose that π values are different from site to site and are the same for all the nodes in a same site (Table 2).

As literature lacks a set of heterogenous computing benchmarks, to evaluate the effectiveness of our MEO we have conceived four different scenarios: one with heavy communications among tasks, one with lighter communications, another in which local node loads are present and a final in which the reliability values of two links decrease. To test the behavior of the mapper also as a function of P, two cases have been dealt with: one in which $P = 20$ and another with $P = 60$.

After a tuning phase $\mathcal{N}_{iter}$ has been set to 200,000, $\mathcal{N}$ to 2,000 and q to 0.95. All the tests have been made on a 1.5 GHz Pentium 4. For each problem 20 MEO runs have been performed, differing in the seeds for the random number generator. Each

run takes about 20 seconds for $P = 20$ and 160 seconds for $P = 60$. Henceforth we denote by μ_{Φ_1} and μ_{Φ_2} the best solutions found in terms of lowest maximal resource utilization time and of highest reliability, respectively.

Due to lack of space, we will hereinafter discuss the experiments with $P = 60$ only and will present the extremal solutions explicitly just for the first experiment, whereas the other ones will be reported concisely as a list whose generic component has the format: $\mathscr{S}(\mathscr{P} : \mathscr{T})$ where $\mathscr{S}$ is the site name, $\mathscr{P}$ the number of nodes used in that site and $\mathscr{T}$ the number of there allocated tasks.

Experiment 1. It regards an application of $P = 60$ tasks divided into three groups $\mathscr{G}_1$, $\mathscr{G}_2$ and $\mathscr{G}_3$ of 20 tasks each. Each task in $\mathscr{G}_1$ has $\gamma_k = 90{,}000$ Mega Instrictions (MI), communicates only with all those in $\mathscr{G}_3$ with $\psi_{km} = 300$ Mbit $\forall m \in [41,\ldots,60]$. Each one in $\mathscr{G}_2$ has $\gamma_k = 900$ MI, communicates only with all those in $\mathscr{G}_3$ with $\psi_{km} = 1{,}000$ Mbit $\forall m \in [41,\ldots,60]$. Each one in $\mathscr{G}_3$ has $\gamma_k = 90$ MI, communicates also with all the others in the same group with $\psi_{km} = 3{,}000$ Mbit $\forall m \in [41,\ldots,60]$. Moreover $\ell_i(\Delta t) = 0.0$ for all the nodes and $\lambda_w = 1.0\ \forall w \in \{A, B.\ldots, J\}$. This problem is quite balanced between computation and communication, and, when use of resources in considered, MEO should find an extremal solution distributing all the tasks in nodes belonging to the sites showing a good balance between these two features, i.e. those of D, E, F and G. Nodes of F should be preferred, instead, when reliability is dealt with. Any execution of the mapper finds out several solutions, all non–dominated in the final Pareto front. The best mappings provided by MEO are:

$$\begin{aligned}\mu_{\Phi_1} = \{&76,95,92,82,88,98,84,91,79,103,77,73,72,91,72,78,100,74,73,85,\\ &73,90,91,80,93,94,82,77,79,93,77,101,102,81,90,86,75,92,87,79,\\ &86,102,99,87,101,95,94,74,88,81,98,80,93,78,75,76,89,83,97,103\}\end{aligned}$$

with $\Phi_1 = 349.706s, \Phi_2 = 0.297$ and

$$\begin{aligned}\mu_{\Phi_2} = \{&113,108,108,104,115,110,117,110,114,115,104,122,116,123,104,\\ &112,117,117,123,116,113,118,109,104,114,123,115,110,110,112,\\ &106,104,115,123,120,108,106,119,115,110,123,116,114,106,106,\\ &115,111,114,107,118,118,110,119,112,107,118,105,114,116,121\}\end{aligned}$$

with $\Phi_1 = 375.0s, \Phi_2 = 0.547$. The presence of sustained computations for some tasks and heavy communications for some others has a strong influence on the extremal solution related to use of resources, which selects all the nodes in the site E. This site has a good balance among number of nodes, their computing power and intrasite bandwidth. In μ_{Φ_2} only nodes in F, the most reliable ones in the whole grid, have been chosen, and loaded with more than one task each.

Experiment 2. It is like the former one but we have decreased the weight of many communications. Namely, each task in $\mathscr{G}_1$ communicates with all tasks in $\mathscr{G}_3$ with $\psi_{km} = 10$ Mbit $\forall m \in [41,\ldots,60]$, and each one in $\mathscr{G}_2$ communicates only with all tasks in $\mathscr{G}_3$ with $\psi_{km} = 100$ Mbit $\forall m \in [41,\ldots,60]$. Communication within group $\mathscr{G}_3$ are the same as in Experiment 1. The best solutions are $\mu_{\Phi_1} = \{A(13 :$

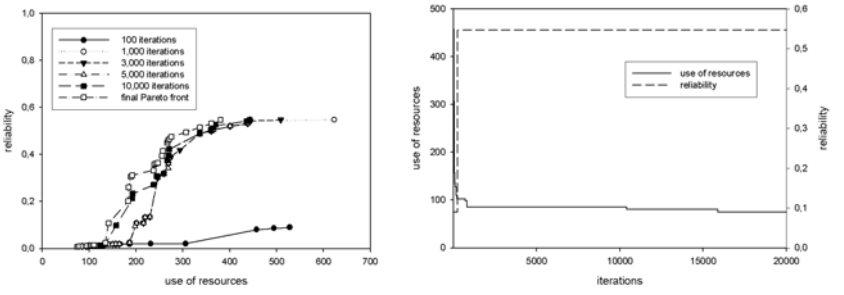


Fig. 2 The final Pareto front achieved for run 1 of Exp. 2 and its evolution (left). Evolution in the same run for the best values of use of resources and reliability (right)

$16); B(3:3); C(1:1); D(1:1); I(16:39)\}$ with $\Phi_1 = 74.764s, \Phi_2 = 0.007$ and $\mu_{\Phi_2} = \{F(19:60)\}$ with $\Phi_1 = 434.63s, \Phi_2 = 0.547$. In μ_{Φ_1} only nodes in the most powerful sites A, B, C and D have been selected for the 20 computation–intensive tasks, and those in I for the ones with higher communications. These latter have been preferred to those in J, which have equal bandwidths, because they have higher α values. Instead, no changes appear for the reliability, since only nodes from F are chosen, as in the previous experiment.

Furthermore, the system proposes other non–dominated solutions better balanced in terms of the two goals. Figure 2(left) shows the final Pareto front achieved in the first of these runs. In it the extremal points on the front represent the two solutions reported above, which are optimal from the point of view of one objective. The theoretically optimal point is $(0.0, 1.0)$, i.e. the one in the top–left corner of the figure. All the solutions belonging to it are provided as an output of MEO tool. Those of them in the middle part of the front are better balanced and users might prefer them to the extremal ones. Moreover, the same figure shows the evolution of the Pareto front for this run. The extremal solutions are in most of the runs found within $20,000$ evaluations. As the number of iterations increases, the front tends to approach the theoretically best solution, i.e. moves upwards and towards left. Moreover, Fig. 2(right) reports the best values for use of resources and for reliability as a function of the number of iterations.

Experiment 3. It is like the second one but we have taken into account the node loads as well. We have set $\ell(\Delta t)$ equal to 0.9 for the eight nodes $[0-7]$ in site A, and to 0.8 for all those in I. We expect that for resource use nodes from I, now heavily loaded and with a very small amount of available power fraction, will not be chosen. This time we have $\mu_{\Phi_1} = \{A(10:10); B(8:8); C(2:2); J(18:40)\}$ with $\Phi_1 = 70.360s, \Phi_2 = 0.003$ and $\mu_{\Phi_2} = \{F(20:60)\}$ with $\Phi_1 = 387.508s, \Phi_2 = 0.547$. In μ_{Φ_1} the most computation–bound tasks use some of the most powerful nodes from A (but none of the eight loaded ones), B and C, while all those with higher communications have been placed onto nodes in J, as good as those in I for the communication, but now with a higher power fraction. In μ_{Φ_2}, as before, all tasks are loaded onto nodes belonging to F.

Table 3 Experimental results

exp. N.	1	2	3	4	5	6	7	8
Φ_1^b	349.70	74.76	70.36	70.36	86.25	31.87	31.87	31.87
			see text		{A(7:7);I(9:13)}	{A(8:8);I(1:1);J(9:11)}	{A(7:7);J(8:13)}	{A(7:7);J(8:13)}
Φ_2^*	0.297	0.007	0.003	0.003	0.186	0.208	0.208	0.207
$\langle\Phi_1\rangle$	378.186	75.99	77.223	76.459	86.25	31.875	31.875	31.87
σ_{Φ_1}	11.173	1.665	2.914	2.443	0.0	0.0	0.0	0.0
Φ_2^b	0.547	0.547	0.547	0.160	0.817	0.817	0.817	0.543
			see text		{F(13:20)}	{F(13:20)}	{F(14:20)}	{G(13:20)}
Φ_1^*	375.00	380.39	387.50	567.08	182.25	180.07	180.07	225.06
$\langle\Phi_2\rangle$	0.547	0.547	0.547	0.160	0.817	0.817	0.817	0.543
σ_{Φ_2}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Experiment 4. The scenario is the same as in the previous case, but we have decreased the reliability of the links related to the sites E and F, which fall down to $\lambda_E = \lambda_F = 0.95$. We expect that these changes yield a μ_{Φ_2} which no longer contains nodes from F. The best solutions found are $\mu_{\Phi_1} = \{A(10:10);B(8:8);C(2:2);J(18:40)\}$ with $\Phi_1 = 70.360s, \Phi_2 = 0.003$ and $\mu_{\Phi_2} = \{G(16:60)\}$ with $\Phi_1 = 557.085s, \Phi_2 = 0.160$. Now μ_{Φ_2} contains only nodes from G, showing the shift from the now worse connected site F to a better connected one, as hoped. Instead, μ_{Φ_1} is the same an in Experiment 3.

Besides the results for $P = 60$, also the findings and the best extremal solutions in concise form for $P = 20$ (Experiments 5 to 8) are reported in Table 3. In it for any test Φ_1^b and Φ_2^b denote the best fitness values for the two objectives, while Φ_2^* and Φ_1^* represent the corresponding values for the other objective. For any objective, also the average $\langle\Phi\rangle$ and the variance σ_Φ computed over the 20 runs are shown. Table 3 evidences that the experiments $5-8$ with $P = 20$ have a variance of 0.0, showing MEO robustness with respect to the initial random seeds. The tests $1-4$ with $P = 60$ show very good robustness when reliability is accounted for, and a non–zero variance when use of resources is considered.

5 Conclusions

In this paper a multisite multiobjective tool based on Extremal Optimization has been proposed to map parallel applications on grid systems. To consider at the same time two objectives, i.e. minimal use of grid resources and maximal hardware reliability, a version based on Pareto dominance has been developed.

The proposed mapper has been tested on eight experiments differing in number of tasks making up the application, weight of communications, load of the grid nodes and faults in site links. These cases represent a set of typical real–time situations, and in all of them the tool has provided the expected solutions.

MEO has two very interesting features when compared to other multiobjective mappers, also implemented by us [16], based on Differential Evolution, the results of which are not shown here due to lack of space. The first feature consists in its higher speed [16], and the second in its ability to provide in the Pareto set also the theoretically best extremal solutions (especially those for the use of resources), which are sometimes missed by the other mappers.

References

1. Mateescu, G.: Quality of service on the grid via metascheduling with resource co–scheduling and co–reservation. Int. Jour. of High Performance Comp. Appl. 17(3), 209–218 (2003)
2. Foster, I.: Globus toolkit version 4: Software for service–oriented systems. In: Jin, H., Reed, D., Jiang, W. (eds.) NPC 2005. LNCS, vol. 3779, pp. 2–13. Springer, Heidelberg (2005)
3. Fernandez–Baca, D.: Allocating modules to processors in a distributed system. IEEE Transactions on Software Engineering 15(11), 1427–1436 (1989)
4. Braun, T.D., Siegel, H.J., Beck, N., Bölöni, L.L., Maheswaran, M., Reuther, A.I., Robertson, J.P., Theys, M.D., Yao, B.: A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. Journal of Parallel and Distributed Computing 61, 810–837 (2001)
5. Kim, S., Weissman, J.B.: A genetic algorithm based approach for scheduling decomposable data grid applications. In: Int. Conf. on Parallel Processing (ICPP 2004), Montreal, Canada, pp. 406–413 (2004)
6. Song, S., Kwok, Y.K., Hwang, K.: Security–driven heuristics and a fast genetic algorithm for trusted grid job scheduling. In: IPDP 2005, Denver, Colorado (2005)
7. Boettcher, S., Percus, A.G.: Extremal optimization: an evolutionary local–search algorithm. In: Bhargava, H.M., Ye, N. (eds.) Computational Modeling and Problem Solving in the Networked World. Kluwer, Boston (2003)
8. Boettcher, S.: Extremal optimization of graph partitioning at the percolation threshold. Journal of Physics A: Mathematical and General 32, 5201–5211 (1999)
9. Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation 3(1), 1–16 (1995)
10. Bak, P., Sneppen, K.: Punctuated equilibrium and criticality in a simple model of evolution. Physics Review Letters 71, 4083–4086 (1993)
11. Dong, F., Akl, S.G.: Scheduling algorithms for grid computing: State of the art and open problems. Technical Report 2006–504, School of Computing, Queen's University Kingston, Ontario, Canada (2006)
12. Fitzgerald, S., Foster, I., Kesselman, C., von Laszewski, G., Smith, W., Tuecke, S.: A directory service for configuring high–performance distributed computations. In: Sixth Symp. on High Performance Distributed Computing, Portland, OR, USA, pp. 365–375. IEEE Comp. Soc., Los Alamitos (1997)
13. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid information services for distributed resource sharing. In: Tenth Symp. on High Performance Distributed Computing, San Francisco, CA, USA, pp. 181–194. IEEE Comp. Soc., Los Alamitos (2001)
14. Wolski, R., Spring, N., Hayes, J.: The network weather service: a distributed resource performance forecasting service for metacomputing. Future Generation Comp. Sys. 15(5-6), 757–768 (1999)

15. Gong, L., Sun, X.H., Waston, E.: Performance modeling and prediction of non–dedicated network computing. IEEE Trans. on Computer 51(9), 1041–1055 (2002)
16. De Falco, I., Della Cioppa, A., Maisto, D., Scafuri, U., Tarantino, E.: Multisite mapping onto grid environments using a multi-objective differential evolution. In: Differential Evolution: Fundamentals and Applications in Engineering, ch. 11. John Wiley, Chichester (to appear)

Interactive Incorporation of User Preferences in Multiobjective Evolutionary Algorithms

Johannes Krettek, Jan Braun, Frank Hoffmann, and Torsten Bertram

Abstract. This paper proposes a novel interactive scheme to incorporate user preferences into evolutionary multiobjective optimization. The approach combines an evolutionary algorithm with an instance based supervised online learning scheme for user preference modeling. The user is queried to compare pairs of prototype solutions in terms of comparability and quality. The user decisions form a preference model which induces a ranking on the population. The model acts as a critic in the selection among non-dominated solutions on behalf of the expert. The user preference is extrapolated from a minimal number of pairwise comparisons to minimize the burden of interactive expert decisions. The preference model includes the concept of comparability to allow simultaneous convergence to multiple disconnected regions of the Pareto front. The preference model comprehends the specific preference scenarios of scalar optimization, goal oriented scenarios, ranking of criteria and global approximation of the Pareto front. It thus represents a general scheme for interactive optimization that does not depend on prior assumptions on either the problem or user preference structure.

1 Introduction

The issues of search and decision making are inherently intertwined in multiobjective optimization. The goal of evolutionary search is to generate a representative set of Pareto optimal solutions, out of which the decision maker selects the solution that best matches his preferences. A posteriori preference articulation works well for multiobjective problems with few criteria in which the evolved discrete population of non-dominated solutions is a suitable approximation of the

Johannes Krettek · Jan Braun · Frank Hoffmann · Torsten Bertram
Chair for Control and Systems Engineering
Technische Universität Dortmund, 44221 Dortmund
e-mail: johannes.krettek@tu-dortmund.de
http://www-rst.e-technik.tu-dortmund.de

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 379–388.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

actual Pareto front. In a priori preference articulation the decision maker specifies his utility function in advance, that transformes the multiobjective problem into a scalar problem. For most practical problems such an approach is unfeasible, as the decision maker is usually unable to specify global trade-offs among conflicting objectives in particular as he lacks knowledge of feasible alternative solutions prior to the search. The handling of preferences in multiobjective evolutionary optimization and the related topic of multicriteria decision making is addressed by C. Coello in [1]. A more recent survey on preference incorporation and interaction is given by Rachmawati et al. in [8]. As a remedy to a priori or a posteriori decision making progressive preference articulation interleaves search and decision making in that the decision maker expresses his preferences among current non-dominated solutions and the trade-offs among the objectives are adjusted accordingly in future search. The decision process is easier than in a priori preference articulation as the decision maker only compares alternative solutions rather than to specify an absolute quantitative utility of solutions. Evaluating every candidate solution is practically unfeasible as the large number of selection decisions in evolutionary optimization imposes an unacceptable burden on the decision maker. Therefore, the decision maker only articulates his preferences on a representative subset of non-dominated solutions, from which a global preference model is inferred. Examples of this approach include probabilistic trade-off development (PROTRADE) [5], progressive articulation of preferences (STEP) [2] and sequential multiobjective problem solving (SEMOPS) [6]. An interactive tool for choosing solutions from evolutionary optimization is presented by Deb et al. in [3]. Our approach to progressive preference articulation is novel as it does not assume a specific model of user preferences, but rather follows the instance based learning paradigm. The preference relation among arbitrary solutions is inferred from a set of training solution pairs for which the decision maker explicitly states his preferences. Our scheme also accounts for the non-comparability of solutions, a concept that is not captured by preference of one solution over another.

2 Interactive Preference Articulation

It is difficult for an expert to quantify preferences and trade-offs among multiple objectives in particular when lacking knowledge about feasible alternative solutions. It is much easier for a decision maker to articulate his preferences during the optimization in cognizance of alternative compromise solutions. Interactive preference articulation raises a number of questions, that have been previously addressed in numerous publications on decision making in multiobjective evolutionary optimization [7, 9]:

1. When and which prototype solutions are presented to the decision maker?
2. Which decision does the user take?

3. How does the user decision effect the selection in future evolutionary optimization?

2.1 Pairwise Solution Comparison Scheme

Ideally the expert states his mutual preference or ranking among all solutions of the current population. In this case selection exactly mimics the experts true utility function rather than an approximation of that function. However, such a complete interaction exceeds the human capability of data processing even in modest optimization problems. The set of non-dominated solutions typically contains many more alternatives than the expert is willing to evaluate. Therefore the interactive decision making is restricted to a reduced set of prototype solutions generated by clustering the current Pareto set.

In our scheme a hierarchical clustering algorithm identifies N clusters that best represent the set of n solutions, of which each is associated to its nearest cluster center. The user expresses his preferences by pairwise comparison of the cluster prototypes in terms of mutual quality and comparability. Solutions $S_i = \{\mathbf{x}_i, f(\mathbf{x}_i)\}$ are represented by their parameter vector $\mathbf{x}_i$ and their criteria vector $f(\mathbf{x}_i)$. The expert compares two solutions S_i, S_j in terms of their mutual preference $\sigma(S_i, S_j) \in [-1, 1]$. In the extreme cases $\sigma(S_i, S_j) = 1$ and $\sigma(S_i, S_j) = -1$ either S_i is totally preferred over S_j or vice versa. The expert deems the two solutions as equal for $\sigma(S_i, S_j) = 0$, whereas any value in between indicates weaker preference for either of the solutions. In addition the expert classifies the degree of comparability $\rho \in [0, 1]$ of two solutions. In case $\rho = 1$ the two solutions are fully comparable. Notice, that there is a fundamental difference between judging two solutions as equal $\sigma = 0$ or as incomparable $\rho = 0$. In the former case the two solutions are subject to a competition in which they are considered equal. In the later case the two solutions are not compared and do not compete with each other. Competition and selection are restricted to subsets of comparable solutions. The set D_σ of pairwise evaluations of prototype solutions $\{S_i, S_j\}$ provide the training instances.

2.2 Preference Estimation Based on Pairwise Similarity

The preference relationship among all individuals in the current generation of the optimization is imposed from the preference model induced by the training data set D_σ. The preference model relies on a nearest neighbor scheme, in which the prediction is based on the similarity of the query with instances in D_σ. The similarity of the query pair $\{P_k, P_l\}$ with an instance $\{S_i, S_j\}$ in D_σ is computed based on their distance in the normalized objective space given by

$$\Delta(\{P_k, P_l\}, \{S_i, S_j\}) = ||f(\mathbf{x}_k) - f(\mathbf{x}_i)|| + ||f(\mathbf{x}_l) - f(\mathbf{x}_j)|| \quad (1)$$

The similarity weight of a training pair is determined by a distance based Gaussian kernel

$$\mathrm{w}(\{P_k,P_l\},\{S_i,S_j\}) = \exp\left(-\frac{\Delta(\{P_k,P_l\},\{S_i,S_j\})}{d_{min}}\right) \tag{2}$$

in which d_{min} denotes the mean minimal distance of the members of the current population in objective space.

The estimated preference of the query pair is computed by the similarity weighted average preference relation of training pairs

$$\hat{\sigma}(P_k,P_l) = \frac{\sum_{i,j}\sigma(S_i,S_j)\mathrm{w}(\{P_k,P_l\},\{S_i,S_j\})}{\sum_{i,j}\mathrm{w}(\{P_k,P_l\},\{S_i,S_j\})} \tag{3}$$

The comparability $\hat{\rho}$ of each pair is estimated in a similar manner:

$$\hat{\rho}(P_k,P_l) = \frac{\sum_{i,j}(\rho(S_i,S_j)-1)\,\mathrm{w}(\{P_k,P_l\},\{S_i,S_j\})}{\sum_{i,j}\mathrm{w}(\{P_k,P_l\},\{S_i,S_j\})} \tag{4}$$

2.3 Preference Controlled Selection Mechanism

The instance based preference model predicts the pairwise preference and comparability of each solution with the $n-1$ other members of the set of non-dominated solutions. The solutions are ranked according to their estimated preference and comparability based on their relative performance index defined by

$$\gamma_\sigma(P_k) = \frac{\sum_{l=1}^{n}\hat{\sigma}(P_k,P_l)\hat{\rho}(P_k,P_l)}{\sum_{l=1}^{n}\hat{\rho}(P_k,P_l)} \tag{5}$$

The performance index captures the average preference of a solution in the context of its comparable competitors. Rather than to strive for a global preference order, the comparability $\hat{\rho}$ restricts competition to more or less disjunct subsets of similar solutions. In addition the comparability index

$$\gamma_\rho(P_k) = \sum_{l=1}^{n}\hat{\rho}(P_k,P_l) \tag{6}$$

captures the relative density of solutions in the preference space. To support an even distribution across the Pareto front the comparability index should be minimized. Notice, the similarity with crowding schemes in which similar solutions are downgraded. The objective is to simultaneously maximize γ_ρ and minimize γ_σ in order to balance exploration and exploitation. The best solutions are selected according to a standard multiobjective selection scheme proposed by Fonseca [4], in which the rank of an individual depends on the number of dominating solutions. The μ best ranked non-dominated solutions with respect to preference and comparability are selected as parents.

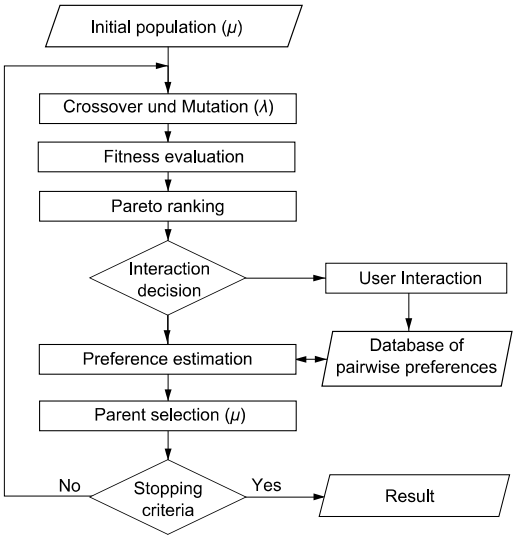


Fig. 1 Evolutionary Algorithm with user preference interaction

2.4 Artificial Benchmark Preference Model

It would be problematic to rely on a human decision maker for a thorough evaluation and analysis of the proposed interactive optimization algorithm. First of all the burden of interaction would be substantial and it is questionable that a human is able to take consistent and reproducible decision over a long time span. In order to overcome this limitation, the human user is replaced with an artificial, transparent user decision model for the purpose of analysis. This scheme allows the objective evaluation of the method in reproducible scenarios at different levels of expert engagement. The artificial user preference captures the proximity of a solution to either a single or multiple hypothetical optimal targets in the normalized objective space. The hypothetical user targets R lie close to the Pareto front albeit in the unfeasible region of the objective space. The mutual preference for a solution pair $\{S_i, S_j\}$ is

$$\sigma_M(S_i, S_j) = \frac{||f(\mathbf{x}_j) - R|| - ||f(\mathbf{x}_i) - R||}{||f(\mathbf{x}_i) - R|| + ||f(\mathbf{x}_j) - R||} \tag{7}$$

The relative preference depends on the relative distance of the two candidate solutions to the target R, such that the solution closer to the target R is preferred over the remote one.

In case of multiple targets, only solution pair of solutions with the same nearest target obtain a mutual preference according to equation 7, whereas pairs belonging to different targets are deemed incomparable ($\rho = 0$). Figures 2 compares the nominal user preference model with the approximated model induced by a sample of

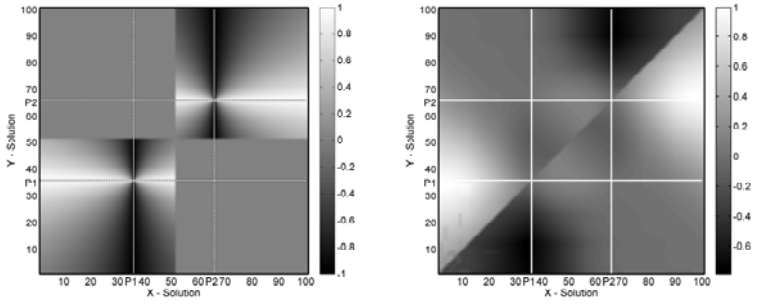


Fig. 2 Preference relation of solution pairs (X,Y) according to the nominal user preference model $\sigma_M(X,Y)$(left) and the approximated model $\hat{\sigma}(X,Y)$ induced by ten equidistant samples (right)

ten equidistant pairwise preference articulations. For the purpose of illustration the objective space is one-dimensional, with two targets $P_1 = 35$ and $P_2 = 65$. The plots illustrate that the nominal mutual preference $\sigma(X,Y)$ (left) and the approximated preference $\hat{\sigma}(X,Y)$ (right) match each other, but that discontinuities of the original model are smoothed out due to the weighted averaging of discrete samples with a Gaussian kernel. The comparability $\hat{\rho}(X,Y)$ shows a similar behavior.

The accuracy of the induced model quality largely depends on the number of training samples, which in turn is determined by the number of clusters c and thereby queries per interaction and the frequency of user interactions, namely the number of elapsed generations n between consecutive user interactions. In contrast to regression models the absolute model error is less relevant. What is more important in the context of interactive evolutionary optimization is the impact of the approximated user preferences on the selection decisions. The selection error is defined as the ratio of parents incorrectly selected according to the approximated preferences, which deviate from the ideal decision according to the nominal user preferences.

Table 1 shows the relationship between the number of queries and frequency of interaction with the false selection rate during the evolution of $\mu = 25$ parents and $\lambda = 100$ offsprings over the course of 50 generations. The reported false selection rates are based on the average of 10 optimization runs. The results demonstrate

Table 1 False selection rate in per cent

	$c = 100$	$c = 25$	$c = 10$	$c = 5$
$n = 1$	2.98	3.34	4.18	5.24
$n = 5$	4.09	4.91	5.39	6.68
$n = 10$	4.90	5.75	5.98	6.92

that a substantial reduction in queries as well as interaction frequencies only causes a moderate deterioration in selection quality. The lower selection quality is easily compensated by a moderate increase in computational effort of the evolutionary

algorithm. In most optimization scenarios the actual runtime of the interactive evolutionary algorithm is determined by the periods of interaction with the user. The number of queries is further reduced by focusing on those pairwise comparisons that provide the largest information. Similar to the concept of active learning in machine learning, in which the learner decides about the next the query, those solution pairs are presented to the decision which explicit preference evaluation has the largest impact on the induced ranking.

3 Results

The convergence behavior of the proposed multiobjective evolutionary algorithm is analyzed with respect to the Kursawe test function

$$f_1(\mathbf{x}) = \sum_{i=1}^{5} \left(-10 \exp\left(-0.2 \sqrt{x_i^2 + x_{i+1}^2} \right) \right) \qquad f_2(\mathbf{x}) = \sum_{i=1}^{6} \left(|x_i|^{0.8} + 5 \sin(x_i)^3 \right) \tag{8}$$

with two objectives f_1 and f_2. The hypothetical user preference model is based upon a single target $P_{Ref} = (-35; -21.5)$. The convergence behavior of the interactive multiobjective optimization is compared with a global multiobjective scheme without user interaction in which selection is based on dominance ranking and niching mechanism only. It is also compared with a scalar evolutionary optimization for which the weighted scalar fitness function $f_g = f_1(x) + 1.5 f_2(x)$ assumes its minimum at the target P_{Ref}. It is expected that the interactive scheme is not as efficient as the ideal scalar optimization with a priori known trade-offs but outperforms the unbiased multiobjective convergence to the global Pareto front. The convergence behavior is analyzed in terms of the minimal distance between the target and the closest solution in the population. Notice, that the scalar optimization directly minimizes this distance as the fitness function f_g is monotonic with the distance to the target. The multiobjective scheme has no notion of proximity to the target as all non-dominated solutions are considered equal. The interactive scheme has no a priori knowledge of the target, but with progressive interaction forms a preference model that captures the proximity to the target.

To visualize the convergence behavior of the proposed variants of evolutionary algorithms, a population of $\lambda = 50$ with $\mu = 10$ parent individuals is evolved over 20 generations. Figure 3 illustrate the evolution of the population as it progresses toward the Pareto front. The gray level of the solutions indicates the generation changing from light gray for the first generations to black for the final generations. The scalar optimization with aggregated objectives shown in figure 3 leads converges rapidly toward the Pareto region in the vicinity of the target $P_{Ref} = (-35; -21.5)$. In case of a non-convex Pareto front, weighted aggregation might introduce multiple local minima of the scalar objective function.

The multiobjective optimization without preference distinction among non-dominated solutions progresses toward the global Pareto front as shown in figure 3. As there is no notion of a target, the density of solutions along the Pareto

front is uniform. Naturally, the convergence toward the target proceeds at a slower rate compared to the scalar and preference based schemes.

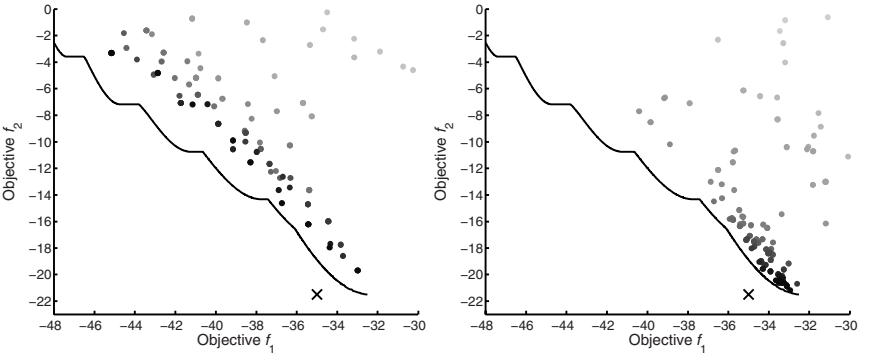


Fig. 3 Comparison of optimization algorithms: multiobjective optimization (left) and scalar optimization (right)

The incorporation of preferences into the multiobjective optimization scheme results in a faster convergence toward the target as shown in figure 4. Notice, that the evolutionary algorithm gradually infers from the pairwise comparisons on the underlying objective. The population evolves toward the target, albeit initially at a slower rate. The preference model becomes more accurate with increasing number of comparisons such that eventually the distance to the target becomes the same as in the scalar case. Figure 4 (right) illustrates the convergence of the current best solution toward the target over the course of evolution. The results are based on the average progress over 25 separate runs of the evolutionary algorithm.

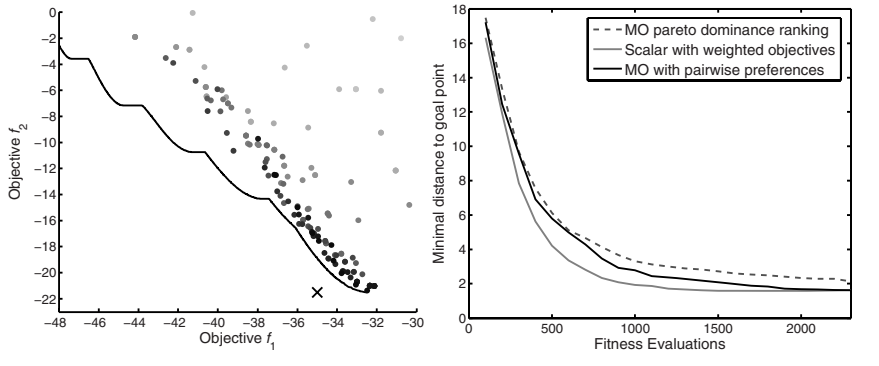


Fig. 4 Preference guided optimization (left), Convergence comparison for different optimization approaches (right)

3.1 Interactive Optimization with Multiple Targets

The concept of incomparable enables the proposed scheme to focus the search on multiple disconnected regions of the criteria space. This section analyzes the convergence behavior for the Kursawe test function with two separate unfeasible targets. The artificial user model has a preference for solutions closer to either of the targets with solutions that belong to different targets deemed as incomparable. The performance of the standard MOEA, a scalar evolutionary optimization and the proposed scheme with preference based selection is compared for an evolutionary algorithm with $\mu = 20$ parents and $\lambda = 100$ offsprings. Figure 5 (left) shows the evolution of the population over the course of optimization with two targets at $P_{Ref1} = (-53; -21)$ and $P_{Ref2} = (-60; -13)$.

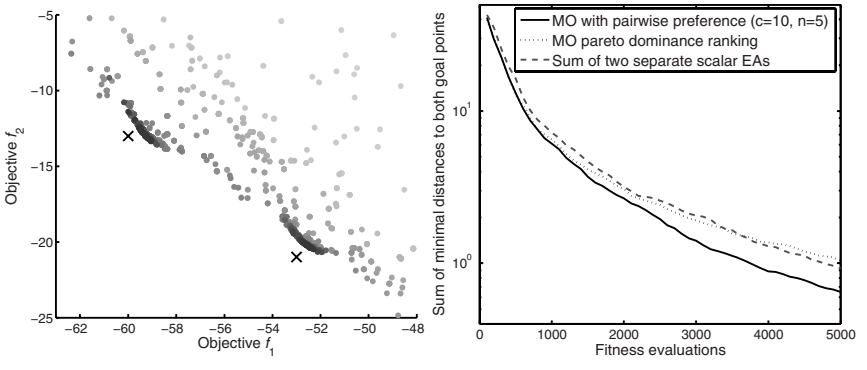


Fig. 5 Evolution of population for two targets (left), convergence comparison for optimization of multiple targets (right)

3.1.1 Convergence Comparison

Figure 5 (right) compares the convergence of the three variants in terms of the minimal distances of the best solutions to the targets. Notice, that the performance evaluates the proximity to both targets, convergence to a single target alone is severely penalized. The results are based on the average of 25 statistically independent runs. In case of the scalar optimization the results are based on the aggregation of two independent runs with separate weighted aggregated fitness for each target for a fair comparison. The performance is evaluated with respect to the absolute number of fitness evaluations, which for the scalar case are evenly shared among the two runs. The MOEA with preference incorporation not only outperforms the standard MOEA but is also more efficient than two separate runs of the scalar optimization with weighted aggregation. The results clearly indicate the ability of our proposed scheme to efficiently approximate the Pareto front in regions of particular interest to the expert. Even in case of a two objective problem, the benefits of guiding the search by expert interaction become apparent. Once the preference model has been

established the convergence behavior of the evolutionary algorithm is similar to the case of an explicit scalar formulation of the expert objectives. Unlike weighted aggregation the scheme is still truly multiobjective in that is not restricted to a single target.

4 Conclusion

This contribution proposes a novel scheme for user preference incorporation in multiobjective evolutionary algorithms. The preference modeled is trained from pairwise comparisons of prototype solutions.The instance based learning scheme makes no prior assumption about the structure of the user preferences and captures the entire spectrum from scalar, multiple targets to purely dominance based optimization objectives. The burden of expert interaction is limited by posing the most informative queries and adopt the frequency of interactions to the quality of the preference model. The results on the single and multiple target scenarios indicate the ability of our scheme to efficiently explore and focus the search on the regions of interest to the expert. The main advantages of our scheme are its general applicability and the tight integration of decision making and optimization. The expert establishes her preference in light of the feasible alternatives rather than specifying her trade-offs a priori.

References

1. Coello, C.A.C.: Handling preferences in evolutionary multiobjective optimization: A survey. In: Proceedings of the CEC 2000, pp. 30–37 (2000)
2. Cohon, J.L.: Multiobjective Programming and Planning. Dover Publications (2004)
3. Deb, K., Chaudhuri, S.: I-emo: An interactive evolutionary multi-objective optimization tool. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 690–695. Springer, Heidelberg (2005)
4. Fonseca, C., Fleming, P.: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: Proceedings of the 5th International Conference on Genetic Algorithms, pp. 416–423 (1993)
5. Goicoechea, A., Duckstein, L., Fogel, M.: Multiobjective programing in watershed management: A study of the charleston watershed. Water Resources Research (12), 1085–1092 (1976)
6. Monarchi, D., Kisiel, C., Duckstein, L.: Interactive multiobjective programming in water resources: a case study. Water Resources Research (9), 837–850 (1973)
7. Parmee, I.C., Cvetkovic, D., Watson, A., Bonham, C.: Multiobjective satisfaction within an interactive evolutionary design environment. Evolutionary Computation 8(2), 197–222 (2000)
8. Rachmawati, L., Srinivasan, D.: Preference incorporation in multi-objective evolutionary algorithms: A survey. In: Proceedings of the CEC 2006, pp. 962–968 (2006)
9. Takagi, H.: Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation. Proceedings of the IEEE 89(9), 1275–1296 (2001)

Improvement of Quantum Evolutionary Algorithm with a Functional Sized Population

Tayarani Mohammad and Akbarzadeh Toutounchi Mohammad Reza*

Abstract. This paper proposes a dynamic structured interaction among members of population in a Quantum Evolutionary Algorithms (QEA). The structured population is allowed to expand/collapse based on a functional population size and partial re-initialization of new members in the population. Several structures are compared here and the study shows that the best structure for QEA is the cellular structure which can be an efficient architecture for an effective Exploration/Exploitation tradeoff, and the partial re-initialization of the proposed algorithm can improve the diversity of the algorithm. The proposed approach is tested on Knapsack Problem, Trap Problem as well as 14 numerical optimization functions. Experimental results show that the proposed Structure consistently improves the performance of QEA.

1 Introduction

Recently we proposed a ring structure sinusoid sized population for QEA (SRQEA) [1]. In this paper several structures and functions for the population of QEA are investigated to find the best population structure and function for the size of the population.. Size of the population is an effective parameter of the evolutionary algorithms and has a great role on the performance of EAs. Several researches investigate the effect of population size and try to improve the performance of EAs by controlling of the size of the population. A functional sized population GA with a periodic function of saw-tooth function is proposed in [2]. Reference [3] finds the best population size for genetic algorithms. Inspired by the natural features of the variable size of the population, [4] presents an improved genetic algorithm with variable population-size. In [5] an adaptive size for the population is proposed for a novel evolutionary algorithm. Reference [6] proposes a scheme to adjust the population size to provide a balance between exploration and exploitation. To preserve the diversity in the population in QEA, [7] proposes a novel diversity preserving operator for QEA.

Tayarani Mohammad
Azad University of Mashhad, Iran
e-mail: tayarani@ieee.org

Akbarzadeh Toutounchi Mohammad Reza
Ferdowsi University of Mashhad,
Departments of electrical engineering and computer engineering, Iran
e-mail: akbarzadeh@ieee.org

* Akbarzadeh Toutounchi Mohammad Reza is also currently with the departments of electrical engineering and computer engineering at Ferdowsi University of Mashhad, Iran.

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 389 – 398.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

This paper compares several structures for QEA to find the best structure for this algorithm and applies a novel operator that is functional sized population on this structure. Several functions for the population are proposed here to find the best function for the population. This paper is organized as follows: Section 2 finds the best structure for QEA, Section 3 proposes the functional sized population, Section 4 finds the best parameters for the proposed algorithm, Section 5 experiments the proposed algorithm on 14 numerical functions and finally Section 6 concludes the proposed operator.

2 Best Structure for QEA

Structure of evolutionary algorithms has a grate role on the performance and evolution process of the algorithms. This section tries to find the best structure for QEA. After finding the best structure for QEA, FSQEA is proposed that uses the best structure which is found in this section. The structures which are examined in this paper are shown in fig 1. The examined structures are ring, cellular, Btree, cluster, Grid, $K_{m,m}$, Ladder, Crossed ladder, Star, Random_h and the structure which is proposed in [9]. In star structure all the q-individuals are connected to each other and in Random_h in each iteration of algorithm each q-individual is connected to h q-individual randomly. To find the best structure for several examinations are performed on several problems. The problems which are used are Knapsack problem, Trap problem and 14 numerical benchmark functions. The detail of experiments is not analyzed here and only the overall results are discussed. For the dimension of 100, for the Knapsack problem and Trap problem is the cellular structure and the best structure for the numerical functions is the Random structure. For the numerical functions the best structure is Random_2 and Random_4 with 4 best results, after these structures, Random_6 places in third place with 3 best results. To find the best structure among these two structure (Cellular and Random_2) and the original structure of QEA which is proposed in [9] the experiments are performed for several dimensions. Other structures are not examined in this step because of the limitation of time and computational resources. Table 1 shows the experimental results on 19 benchmark functions for six dimensions. For the dimension of m=25, in 4 objective functions the best

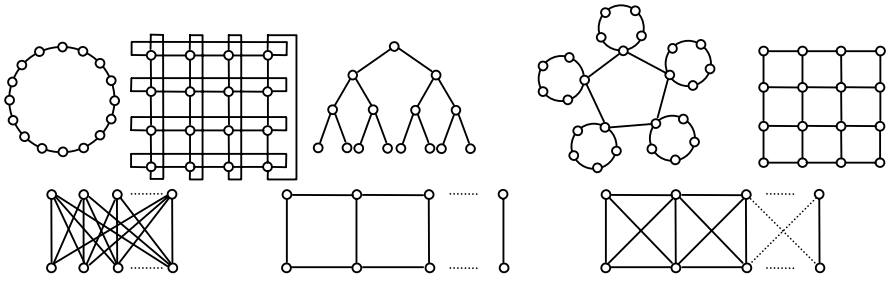


Fig. 1 The compared structures. The structures from top, left to bottom right are Ring, Cellular, Btree, Cluster, Grid, $K_{m,m}$, Ladder and Crossed-ladder

Table 1 The best Structure for QEA. The bold results are the best Ones. All the experiments are performed over 30 runs

	m=25	m=50	m=100	m=150	m=250	m=500	
Reference[9]	4	3	4	6	3	2	%0.19
Cellular	10	14	15	10	13	14	%0.67
Random_2	5	2	0	3	3	3	%0.14

structure is the original structure of QEA, the cellular structure reaches the best results for 10 objective functions and Random_2 structure is the best structure for 5 objective functions. In %67 of the experiments the best structure for QEA is the Cellular structure, in %19 of experiments the best structure is the original structure of QEA and in %14 of experiments the best structure is Random_2. According to these experiments the best structure for QEA is the Cellular structure, so we use the Cellular structure for QEA in the reminder of this paper.

3 Functional Sized Population QEA (FSQEA)

Another approach to maintain the diversity of the population and improve the performance of the evolutionary algorithms is using a variable size for the population. In [1] a variable size population is proposed for QEA that improves the performance of QEA. They use a sinusoid function for the size of the population with partially reinitialization of the q-individuals. Here to improve the performance of QEA, and investigate the best structure and best function for the population of QEA this paper uses a functional population size for QEA. In addition to the sinusoid function, this paper uses some other functions for QEA; the functions are saw-tooth [2], inverse saw-tooth, triangular, sinusoid [1] and square functions. Fig. 1 shows the functions which are examined in this paper. The pseudo code of the proposed Functional Size QEA (FSQEA) is described as below:

Procedure FSQEA
begin
    $t=0$
1. initialize quantum population $Q(0)$ with the size of $n(0) = \overline{n}$
2. make $X(0)$ by observing the states of $Q(0)$.
3. evaluate $X(0)$.
4. **for** all binary solutions x^0_i in $X(t)$ **do**
    begin
5. find neighborhood set N_i in $X(0)$.
6. find binary solution x with best fitness in N_i
7. save x in B_i
    end
8. **while not** termination condition **do**
    begin
    $t=t+1$
9. $n(t) = f(t)$

```
10.   if n(t)>n(t-1) create random q-individuals
11.   if n(t)<n(t-1) eliminate the q-individuals with worst observed fitness
12.   make X(t) by observing the states of Q(t-1)
13.   evaluate X(t)
14.   update Q(t) based on B_i and X(t) using Q-gates
15.   for all binary solutions x^t_i in X(t) do
        begin
16.       find neighborhood set N_i in X(t).
17.       select binary solution x with best fitness in N_i
18.       if x is fitter than B_i save x in B_i
        end
      end
end
```

The pseudo code of FSQEA is described as below:

1. In the initialization step, the quantum-individuals $q^0{}_i$ are located in a structured population. Then $[\alpha_i^0 \ \beta_i^0]^T$ of all $q^0{}_i$ are initialized with $1/\sqrt{2}$, where i=1,2,…,n is the location of the q-individuals in the population, k=1,2,...,m, and m is the number of qubits in the individuals. This implies that each qubit individual $q^0{}_i$ represents the linear superposition of all possible states with equal probability.

2. This step makes a set of binary instants $X(0)=\{x_i^0 | i=1,2,\ldots,n\}$ at generation t=0 by observing $Q(0)=\{q_i^0 | i=1,2,\ldots,n\}$ states, where $X(t)$ at generation t is a random instant of qubit population and n is the size of population. Each binary instant, $x^0{}_i$ of length m, is formed by selecting each bit using the probability of qubit, either $|\alpha_{i,k}{}^0|^2$ or $|\beta_{i,k}{}^0|^2$ of $q^0{}_i$. Observing the binary bit $x^t{}_{i,k}$ from qubit $[\alpha_{i,k}{}^t \ \beta_{i,k}{}^t]^T$ performs as:

$$x_{i,k}^t = \begin{cases} 0 & if \quad \mathrm{R}(0,1) < |\alpha_{i,k}^t|^2 \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

Where $\mathrm{R}(\cdot,\cdot)$ is a uniform random number generator.

3. Each binary instant $x^0{}_i$ is evaluated to give some measure of its objective. In this step, the fitness of all binary solutions of $X(0)$ are evaluated.

4,5,6,7. In these steps the neighborhood set N_i of all binary solutions $x^0{}_i$ in $X(0)$ are found and the best solution among N_i is stored in B_i. In the structured proposed algorithm each individual is the neighbor of itself that is x_i belongs to neighborhood set N_i. B_i is the best possible solution, which the q-individual $q^t{}_i$ has reached.

8. The while loop is terminated when the termination condition is satisfied. Termination condition here is when maximum number of iterations is reached.

9. In the proposed algorithm, the size of the population is a function of the iteration number. In this step, $n(t)$, the size of the population in iteration t, is calculated as a function. The functions that used in this paper are:

Saw-tooth [3]: $n(t) = Round\left[\overline{n} - \frac{2A}{T-1}\left(t - T \times Round\,\frac{t-1}{T} - 1\right)\right]$

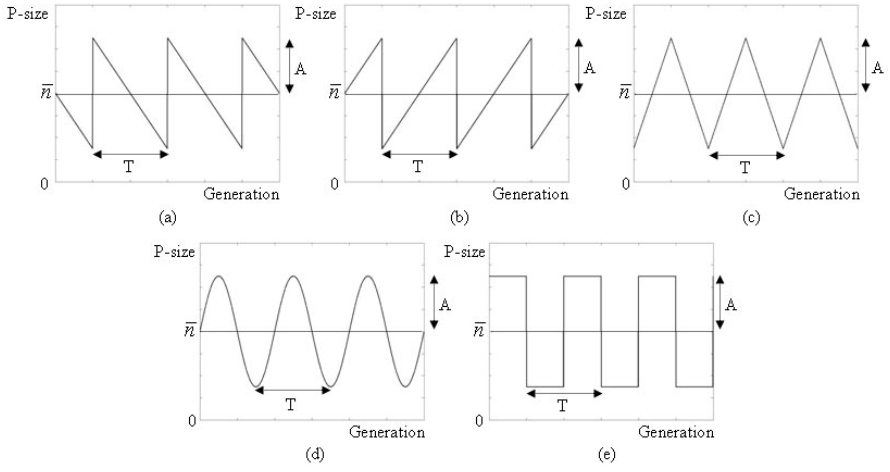


Fig. 2 a) The functions which are used for the population size. a) saw-tooth b) inverse saw-tooth c) triangular d) sinusoid e) square. T is the period of the functions, A is the amplitude and P-size is the size of the population in generation *t*

Inverse Saw tooth: $n(t) = Round\left[\overline{n} + \frac{2A}{T-1}\left(t - T \times Round\,\frac{t-1}{T} - 1\right)\right]$

Triangular: $n(t) = \text{Round}\left[\overline{n} - A + 2A\max\left(\min\left(\frac{2\bmod(t,T)}{T}, 2 - \frac{2\bmod(t,T)}{T}\right), 0\right)\right]$

Sinusoid: $n(t) = Round\left[\overline{n} + A\sin\left(\frac{2\pi}{T}t\right)\right]$

Square: $n(t) = Round\left[\overline{n} + A - 2A \times Round\left(\frac{\bmod(t,T)}{T}\right)\right]$

Where $n(t)$ is the size of the population in generation t, $\overline{n}$ is the average size of the population, A is the amplitude of the periodic function of population size, T is the period of the functional population, Round(.) is the round function (rounds its input to nearest integer), and mod(.,.) is modulus after division function. Fig 3 shows the functions which are used in this paper. The best values for T and A are found in the following of this section.

10. If $n(t)$, the size of the population in iteration t is greater than $n(t\text{-}1)$, it means that the size of the population is increased. So creating random q-individuals, until the size of ring structured population be equal to $n(t)$.

11. If $n(t)$, the size of the population in iteration t is smaller than $n(t\text{-}1)$, eliminate the q-individuals which have the worst observed solution, until the size of ring structured population reaches $n(t)$.

12. Observing the binary solutions $X(t)$ from $Q(t)$.

Table 2 The best parameters for the proposed FSQEA. The best results are the best ones

	Saw-Tooth			Inverse-Saw			Sinusoid			Square			Triangular			QEA
	A	*T*	Best	*A*	*T*	Best	*A*	*T*	Best	*A*	*T*	Best	*A*	*T*	Best	Best
KR1	0.4	25	406.7	0.2	100	407.51	0.4	100	407.23	0.2	250	**407.67**	0.1	50	407.18	387.74
KR2	0.9	100	412.62	0.4	100	412.8	0.2	50	**413.05**	0.4	500	412.59	0.4	100	412.72	407.43
KP1	0.2	50	**556.69**	0.1	50	556.69	0.2	50	556.69	0.2	25	556.69	0.2	50	556.69	517.66
KP2	0.4	50	406.44	0.4	25	**407.56**	0.2	25	407.19	0.4	250	407.49	0.1	25	405.35	388.88
Trap	0.2	250	82.6	0.4	500	83.7	0.2	500	83.7	0.2	500	**84.4**	0.1	500	84.2	79.737
f_1	0.2	100	44932	0.6	100	47227	0.9	100	**47678**	0.4	100	45464	0.2	100	45973	32471
f_2	0.2	100	-1420	0.4	100	**-1274**	0.2	50	-1398	0.2	100	-1419	0.2	250	-1374	-2281
f_3	0.2	250	-17.07	0.4	100	**-16.88**	0.2	50	-16.98	0.2	25	-17.00	0.4	100	-17.00	-17.24
f_4	0.2	500	-22.85	0.2	250	**-17.44**	0.4	100	-21.22	0.2	250	-21.33	0.4	100	-20.74	-47.744
f_5	0.2	100	-1.0e5	0.4	500	-78259	0.4	500	-94385	0.4	250	**-3.8e4**	0.4	100	-9.0e4	-2.05e5
f_6	0.2	100	-22786	0.4	250	-18903	0.4	250	-21103	0.2	100	-2.2e4	0.4	250	-2.1e4	-49138
f_7	0.2	250	32.38	0.4	250	**35.40**	0.4	100	32.36	0.4	500	32.45	0.2	50	33.33	19.33
f_8	0.2	250	50.17	0.4	250	**53.64**	0.4	250	52.28	0.2	100	50.67	0.2	100	51.36	37.49
f_9	0.2	500	-2.5e5	0.4	250	-1.94e5	0.2	100	-2.27e5	0.2	100	-2.3e5	0.2	100	-2.3e5	-5.69e5
f_{10}	0.2	250	-3.55	0.4	500	**-2.99**	0.2	100	-3.33	0.2	250	-3.38	0.4	100	-3.3591	-5.5741
f_{11}	0.2	25	-162.19	0.2	500	-158.75	0.2	100	-161.56	0.2	250	-159	0.2	500	-163.13	-143.63
f_{12}	0.2	250	-7.1e6	0.4	500	-6.22e6	0.4	500	-7.8e6	0.2	100	-7.7e6	0.4	500	-7.1e6	-2.54e7
f_{13}	0.2	100	-39280	0.4	250	-31939	0.2	500	-37418	0.2	500	-36826	0.4	100	-34912	-1.10e5
f_{14}	0.6	25	-0.0057	0.6	25	-0.004	0.6	25	-0.009	0.9	50	**-0.001**	0.9	25	-0.0058	-1.13

13. Evaluating the binary solutions $X(t)$.
14. The quantum individuals are updated using Q-gate.
15. The "**for**" loop is for all binary solutions x^t_i (i=1,2,…,S) in the population.
16. Finding the neighbors of the binary solution located on the location i.
17. Find the best possible solution in the neighborhood N_i, and store it to x.
18. If x is fitter than B_i, store x to B_i.

The proposed functions for the population have two cycles. One cycle is increasing the size of population. In the increasing cycle, the new quantum individuals are created and inserted in the population. Creating new random quantum individuals increases the diversity of the population and improves the exploration performance of the algorithm. The other cycle is the decreasing cycle. In this cycle, the worst quantum individuals of the population are eliminated. This treatment improves the exploitation of the algorithm by exploiting the best solutions and ignoring the inferior ones. This means that the proposed algorithm has two cycles: exploration cycle and exploitation cycle.

4 Finding the Best Parameters

As it seen in Fig. 1, the proposed functions have some parameters that are A, the amplitude and T the period of the functions. In order to find the best values for these parameters some experiments are performed. Fig. 1 shows the finding of the best parameters for the proposed FSQEA for Knapsack Problem penalty type 1 and Generalized Schwefel Function 2.26. The best parameters for the Knapsack problem, Trap Problem and 14 numerical benchmark functions are found similar

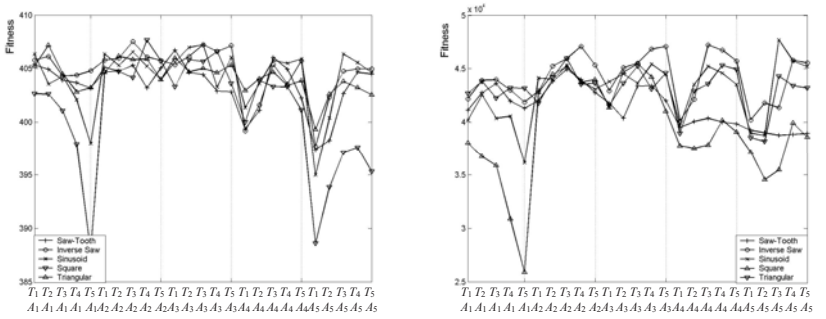


Fig. 3 Parameter setting of FSQEA for T and A for (a) Knapsack Problem Penalty 1 (b) Generalized Schwefel Function 2.26 for several functions for the population. The parameters are set to $T_1 \ldots T_5$=(25,100,250,500,1000) and $A_1 \ldots A_5 = \bar{n} \times$ (0.1,0.2,0.4,0.6,0.9)

to the Fig 2. The best parameters and the best functions for the size of the population are summarized in Table 2. According to Table 2 the Inverse Saw-Tooth function has the best results for 11 benchmark objective functions, the Square for 3 benchmark functions, sinusoid for 2 functions, saw-tooth function for 1 benchmark function, and Triangular with no objective function, so the best function for the size of the population is Inverse Saw-Tooth function. Only for one objective function the best result is reached by original version of QEA and the proposed algorithm improves the performance of QEA for most of the objective functions. In order to find the best parameters for the proposed algorithm Table 3 shows the median and standard deviation of the best parameters for 5 proposed functions. According to this table the best amplitude for the proposed functions in 0.2 and best period T is 100.

Table 3 Median and Standard deviation of the best parameters for the proposed FSQEA

	A		*T*	
	Mean	Std	Mean	Std
Saw-Tooth	0.2	0.184	100	144
Inverse-Saw	0.4	0.124	250	177
Sinusoid	0.2	0.18	100	179
Square	0.2	0.17	250	168
Triangular	0.2	0.18	100	161

5 Experimental Results

The proposed FSQEA is compared with the original version of QEA and DPCQEA[8]. In order to compare the algorithms with their best parameters, the best parameters for each algorithm are found independently. The experimental results are performed for several dimensions (m=25, 50, 100, 250, 500) of Knapsack Problem, Trap Problem and 14 numerical benchmark functions. The

Table 4 Experimental results on 14 numerical benchmark functions for m=100 and m=250. the results are averaged over 30 runs. Ttest shows Ttest between the results of each algorithm with FSQEA. The best results are the best ones

m=100								
	FSQEA		PDCQEA			QEA		
	Mean	Std	Mean	Std	Ttest	Mean	Std	Ttest
KR1	**429.79**	3.66	396.62	12.27	1.76e-7	417.07	6.62	4.74e-5
KR2	414.48	0.48	**434.08**	3.07	1.06e-13	412.51	1.64	0.002
KP1	**562.13**	0.92	519.54	3.19	0	546.82	11.29	0.0004
KP2	**418.15**	1.94	398.61	1.80	6.88e-15	406.91	4.39	7.31e-7
Trap	81.1	0.99	**82**	1.05	0.06	72.2	3.55	4.79e-7
f_1	**45889**	1558.8	38937	2687.6	1.34e-6	34437	3984.1	1.08e-7
f_2	**-1287.8**	97.97	-1684.8	183.4	1.04e-5	-2096.3	199.45	9.89e-10
f_3	**-16.89**	0.14	-16.92	0.12	0.65	-17.19	0.09	2.91e-5
f_4	**-30.08**	5.24	-32.48	3.15	0.23	-39.60	8.48	0.007
f_5	**-1.43e5**	14756	-1.46e5	15450	0.62	-1.67e5	16807	0.004
f_6	**-23017**	2742.1	-32625	4038.4	7.15e-6	-36949	4918.7	3.36e-7
f_7	**31.17**	2.53	22.69	1.40	2.93e-8	22.04	2.46	1.84e-7
f_8	**50.57**	1.75	46.38	3.76	0.005	38.19	3.24	3.58e-9
f_9	**-2.46e5**	23972	-3.62e5	35152	8.79e-8	-4.55e5	68039	3.39e-8
f_{10}	**-4.2161**	0.29	-4.34	0.94	0.67	-5.21	0.68	0.0005
f_{11}	**-169.84**	7.76	-172.19	4.15	0.41	-176.72	3.78	0.02
f_{12}	**-1.33e7**	2.53e6	-1.72e7	2.20e6	0.002	-2.56e7	9.20e6	0.0007
f_{13}	**-49229**	5885	-69515	7918.2	4.09e-6	-1.08e5	35412	5.60e-5
f_{14}	-0.098	0.065	**-0.0005**	0.0005	0.0002	-1.29	1.26	0.008
M=250								
	FSQEA		PDCQEA			QEA		
	Mean	Std	Mean	Std	Ttest	Mean	Std	Ttest
KR1	985.45	15.85	**1003.8**	6.55	0.003	919.75	16.89	4.63e-8
KR2	1010.1	5.03	**1045.1**	8.73	2.02e-9	975.36	18.00	1.47e-5
KP1	1252.6	17.40	**1361.2**	24.44	1.07e-9	1173	30.19	1.02e-6
KP2	994.46	10.31	**183.1**	4.04	8.21e-10	942.99	20.90	1.60e-6
Trap	**198.9**	1.45	183	4.04	8.2e-10	157.4	9.54	6.64e-11
f_1	**76292**	2939	66920	4602	3.71e-5	55844	5845.3	1.07e-8
f_2	**-5118.9**	304.75	-6360	264.68	1.36e-8	-6511	360.74	2.59e-8
f_3	-17.39	0.11	**-17.21**	0.02	7.61e-5	-17.62	0.11	0.0002
f_4	-134.53	10.16	**-132.79**	6.83	0.65	-154.42	14.04	0.002
f_5	-5.60e5	20743	**-5.54e5**	25772	0.58	-6.05e5	43232	0.007
f_6	**-1.09e5**	5039	-1.26e5	3670	4.92e-8	-1.44e5	8269.3	1.05e-9
f_7	**52.24**	4.39	38.38	1.85	3.25e-8	39.43	3.02	5.18e-7
f_8	93.51	5.31	**94.27**	5.95	0.76	73.50	5.02	7.91e-8
f_9	**-1.26e6**	93119	-1.42e6	75573	0.0005	-1.64e6	1.34e5	8.91e-7
f_{10}	**-5.95**	0.30	-5.97	0.36	0.91	-6.52	0.46	0.004
f_{11}	**-189.69**	2.35	-190.31	1.77	0.51	-192.03	1.36	0.014
f_{12}	**-2.44e8**	2.45e7	-2.64e8	2.36e7	0.08	-3.09e8	3.59e7	0.0002
f_{13}	**-3.1e5**	35376	-4.15e5	31888	2.19e-6	-4.81e5	72311	3.21e-6
f_{14}	-7.53	1.94	**-0.11**	0.13	4.77e-10	-20.38	6.19	6.6e-6

average population size of all algorithms for all of the experiments is set to 25; termination condition is set for a maximum of 1000 generations and the structure of population is considered as cellular[8]. Due to statistical nature of the optimization algorithms, all results are averaged over 30 runs and Ttest analysis is performed on results. The parameter of QEA is set to $\Delta\theta$=0.01π (reference [9]

shows this is the best parameter for QEA), the parameters of FSQEA are set to the best parameters found in previous section and the best parameters of DPCQEA are set to the best parameters proposed in [9].

Table.4 summarizes the experimental results on QEA, DPCQEA and FSQEA for Knapsack Problem, Trap Problem and 14 benchmark functions (The results for some dimensions are not summarized in Table.4 because of small space of the paper). As it seen in Table 4, FSQEA has the best results. The Ttest between the results of FSQEA and QEA is too small (in average about 10^{-5}) and it means that the results of Table 4 has high amount of validity.

6 Conclusion

This paper proposes a Functional Sized population QEA with a cellular structure. Before proposing the functional sized QEA, the best structure for QEA is found that is the cellular structure. After finding the best structure for QEA, the FSQEA is proposed with the Cellular structure. The proposed FSQEA has several parameters that are investigated in this paper. Finally the experimental results are performed on the proposed algorithm and the improvement that is shown and the small amount of Ttest shows that the proposed algorithm reaches the best results with high validity. The time complexity of the proposed FSQEA is equal to original version of QEA because the average size of the population for FSQEA is equal to QEA and the number of function evaluations for both of algorithms is equal.

The objective functions which are used here are f_1:Schwefel 2.26 [6], f_2:Rastrigin [6], f_3:Ackley [6], f_4:Griewank [6], f_5:Penalized 1 [6], f_6:Penalized 2 [6], f_7:Michalewicz [7], f_8:Goldberg [2], f_9:Sphere Model [6], f_{10}:Schwefel 2.22 [6], f_{11}:Schwefel 2.21 [6], f_{12}:Dejong [7], f_{13}:Rosenbrock [2], and f_{14}:Kennedy [2].

References

1. Tayarani-N, M.-H., Akbarzadeh-T, M.R.: A Sinusoid Size Ring Structure Quantum Evolutionary Algorithm. In: IEEE International Conference on Cybernetics and Intelligent Systems Robotics, Automation and Mechanics (2008)
2. Koumousis, V.K., Katsaras, C.P.: A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance. IEEE Trans. Evol. Comput. 10, 19–28 (2006)
3. Wang, D.L.: A study on the optimal population size of genetic algorithm. In: Proceedings of the 4th World Congress on Intelligent Control and Automation (2002)
4. Shi, X.H., Wan, L.M., Lee, H.P., Yang, X.W., Wang, L.M., Liang, Y.C.: An improved genetic algorithm with variable population-size and a PSOGA based hybrid evolutionary algorithm. In: International Conference on Machine Learning and Cybernetics (2003)
5. Li-Shan, Q.J.K.: A novel dynamic population based evolutionary algorithm for revised multimodal function optimization problem. In: Fifth World Congress on Intelligent Control and Automation (2004)

6. Zhong, W., Liu, J., Xue, M., Jiao, L.: A Multi-agent Genetic Algorithm for Global Numerical Optimization. IEEE Trans. Sys, Man and Cyber. 34, 1128–1141 (2004)
7. Khorsand, A.-R., Akbarzadeh-T, M.-R.: Quantum Gate Optimization in a Meta-Level Genetic Quantum Algorithm. In: IEEE International Conference on Systems, Man and Cybernetics (2005)
8. Tayarani-H, M.-H., Akbarzadeh-T, M.-R.: A Cellular Structure and Diversity Preserving operator in Quantum Evolutionary Algorithms. In: IEEE World Conference on Computational Intelligence (2008)
9. Han, K., Kim, J.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Trans. on Evolutionary Computation 6(6) (2002)

Appendix

In this section two combinatorial optimization problems, Trap problem and Knapsack problem, and 14 function optimization problems are discussed to evaluate the proposed SRQEA.

Trap problem is defined as:

$$f(x) = \sum_{i=0}^{N-1} \mathrm{Trap}(x_{5i+1}, x_{5i+2}, x_{5i+3}, x_{5i+4}, x_{5i+5}) \tag{2}$$

Where N is the number of traps and

$$\mathrm{Trap}(x) = \begin{cases} 4 - \mathrm{ones}(x), & \text{if } \mathrm{ones}(x) \le 4 \\ 5 & \text{if } \mathrm{ones}(x) = 5 \end{cases} \tag{3}$$

Where the function "ones" returns the number of ones in the binary string x. Trap problem has a local optimum in $(0,0,0,0,0)$ and a global optimum in $(1,1,1,1,1)$.

Knapsack problem is a well-known combinatorial optimization problem which is in class of NP-hard problems [7]. Knapsack problem can be described as selecting various items x_i (i=1,2,…,m) with profits p_i and weights w_i for a knapsack with capacity C. Given a set of m items and a knapsack with capacity C, select a subset of the items to maximize the profit $f(x)$:

$$f(x) = \sum_{i=1}^{m} p_i x_i \quad , \quad \sum_{i=1}^{m} w_i x_i \le C \, .$$

This paper considered:

$w_i = \mathrm{R}(1, v)$, $p_i = \mathrm{R}(1, v)$

Where $\mathrm{R}(\cdot,\cdot)$ is a uniform random number generator and v=10.

The use of QEA for solving Knapsack problem is described in [7].

Optimal Path Planning for Controllability of Switched Linear Systems Using Multi-level Constrained GA

Alireza Rowhanimanesh, Ali Karimpour, and Naser Pariz

Abstract. In this paper, optimal switching signal as well as control input is designed for a general switched linear system using multi-level constrained genetic algorithms (MLCGA). Given any two states in the controllable subspace, the proposed approach automatically finds optimal switching signal and control input which steer the system from initial state to final state in a desired feasible time. From optimization perspective, this problem has several linear constraints such as controllability condition and desired dwell time as well as desired final time. Also, the problem is mixed-variable when the switching indices must be integer. The objective function may be nonlinear, multi-modal and non-analytical. Generally the problem must be solved in two levels. At the bottom level, an optimal control input is found for a candidate switching signal and at the top level an optimizer searches for optimal switching signal. To solve this complex problem, we propose Multi-Level Constrained Genetic Algorithms (MLCGA) which can solve this problem efficiently. As it is demonstrated by a simulation example, using the proposed approach an optimal switching signal with desired dwell time as well as optimal control input in the presence of actuator saturation can be efficiently designed.

1 Introduction

"Hybrid systems" is one of the most recent and hot topics in systems and control theory which attracts researchers of both theoretical and practical domains. A hybrid system includes two distinct types of components, subsystems with continuous dynamics and subsystems with discrete event dynamics that interact with each other. From the perspective of systems and control theory, a hybrid system is considered as continuous systems with switching and a greater emphasis is placed on properties of the continuous state [1]. These systems are called switched systems and stability analysis and control synthesis are the main topics of switched systems theory.

Alireza Rowhanimanesh
Cognitive Computing Lab,
Ferdowsi University of Mashhad, Mashhad, Iran
e-mail: rowhanimanesh@ieee.org

Ali Karimpour · Naser Pariz
Department of Electrical Engineering
Ferdowsi University of Mashhad, Mashhad, Iran
e-mail: karimpor@um.ac.ir, n-pariz@um.ac.ir

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 399 – 408.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

Generally, a switched dynamical system is composed of a family of subsystems and a rule that governs a switching among them (Figure 1). As a result, besides subsystems a switched system also consists of a switching device usually called supervisor [1]. The supervisor produces the switching rule (also called switching signal or switching law) which orchestrates the switching among the subsystems. The dynamics of the switched system is determined by both the subsystems and the switching signal. Switched linear system is a type of switched system when the subsystems are linear (often linear time invariant). Switched linear system is an important and applied field of switched system that large amount of theorems are available to analyze their stability, controllability and etc. One of the most basic concepts in control theory and applications is controllability. There are various definitions for controllability including local and global. Controllability of a switched system is more complicated than an ordinary system with single dynamics. Controllability plays an important role in control of a system. Although several theorems are available for controllability of a switched linear system, these theorems just find reachable and controllable sets. Path planning for controllability is a valuable concept which has been briefly introduced in [1]. Path planning for controllability means finding a feasible switching signal as well as feasible control input which steer the system from desired initial state to desired final state in desired finite time. With respect to the practical domain, there exist several objectives to reach the desired performance. As a result, optimal path planning is hardly required. As it is discussed in part 2, from optimization perspective, optimal path planning for controllability of a general switched linear system is a constrained problem. There are two types of constraints including constraints on switching signal and constraints on controllability input. Fortunately, both of these types of constraints are linear. Although the decision variables of control input design is often continuous, optimal switching signal design often includes mixed (both continuous and discrete) decision variables. Regarding the desired performance in a practical control system design, objective function may be nonlinear, multi-modal and non-analytical. Furthermore, the design process includes two levels. At the bottom level, an optimal control input is found for a candidate switching signal and at the top level an optimizer searches for optimal switching signals. In this paper, we propose a general framework based on multi-level constrained genetic algorithms (MLCGA) to solve this complex optimization problem.

In continue, several previous works by others are briefly considered. Ge et al. (2001) [2], Hihi et al. (2007) [3] and Zhijian Ji et al. (2008) [4] considered theoretical aspects of controllability of switched linear systems. Zongbo Hu et al. (2004) [5] discussed output controllability of switched power converters as switched linear systems. Xie et al. (2004) [6] considered the controllability of periodically switched linear systems with saturating actuators as well as Liang Lv et al. (2007) [7] which designed switched linear system under actuator saturation. Li et al. (2004) [8] designed an optimal switching law for switched linear systems based on convergence route. Dong Chen et al. (2004) [9] used dynamic programming as well as evolutionary computing to optimally control the switched systems. Zhijian Ji (2006) [10] determined the number of switching as well as designing switching sequences for controllability of switched linear systems.

In comparison with the previous works, in this paper we propose a general framework to design optimal switching signal as well as optimal control input for controllability of switched linear systems. Using the proposed approach all practical constraints and objectives including dwell time, actuator saturation and desired final time as well as any nonlinear and non-analytical objective functions can be efficiently considered. In continue we first introduce the mathematical preliminaries of path planning for controllability of a switched linear system in part 2. Next, in part 3, we generally formulate the problem of path planning for controllability. Then we proposed MLCGA to solve this problem. At part 4, a simulation example is designed using the proposed approach to demonstrate the efficiency of this framework.

2 Path Planning for Controllability of Switched Linear Systems

2.1 General Switched Linear System

A general non-autonomous switched linear system is shown in Figure 1. As figure indicates, a supervisor produces switching signal which switches the subsystems. Generally the subsystems are non-autonomous LTI systems. There are several types of switching signals. Time-driven switching law as well as state/output feedback switching law is the commonest one. The path planning procedure in the next subsection uses time-driven switching law. Regarding real world systems, the subsystems can not be switched very quickly. As a matter of fact for any application a time τ exists which for any two consecutive switching times $t_{k+1} - t_k \succ \tau$ must be satisfied. This critical time is called dwell time. Using the proposed approach in this paper, an optimal switching signal is designed with desired dwell time. In the next subsection, without considering the controllability theorems we directly deal with the path planning problem. Readers can refer to [1] to know more details about the controllability of switched linear systems.

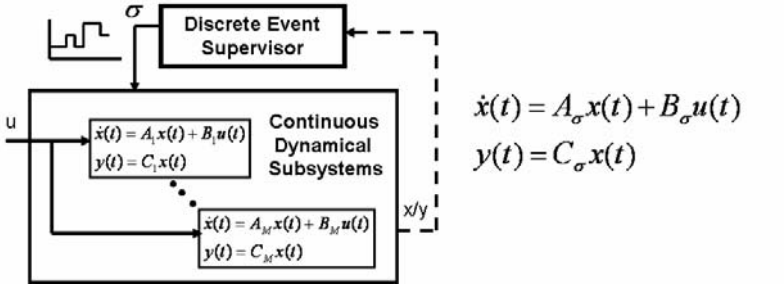


Fig. 1 Switched linear system (left), State space representation (right)

2.2 Path Planning for Controllability

According to [1], given any two states x_0 and x_f in the controllable subspace, path planning means finding feasible switching path σ and control input u to

steer the system from x_0 to x_f in a finite time. Using the theorems of controllability of linear switched system, [1] has formulated a procedure for this path planning. Without dealing with the details, we directly introduce the procedure here.

Suppose $t_0 = 0$. According to [1], we can find a natural number L (the number of switching), positive real numbers $h_0,...,h_L$ (switching durations), and an index sequence $i_0,...,i_L \quad i_k \in \{1,2,...,\mathrm{M}\}$ such that the controllability set requirements are satisfied. It is clear that $t_k = t_{k-1} + h_{k-1} \quad k = 1,...,L+1$. Regarding [1], consider the piecewise continuous control strategy given by $u = B_{i_k}^T e^{A_{i_k}^T (t_{k+1}-t)} a_{k+1}$, $t_k \le t \prec t_{k+1},\ k = 0,...,L+1$ where $a_k \in R^n,\ k = 1,...,L+1$ is a constant vector variables to be determined. Combining the solution of a switched linear system with this control strategy, the following system of equation is achieved as a linear constraint on a:

$$x_f - e^{A_{i_L} h_{i_L}} ... e^{A_{i_0} h_{i_0}} x_0 = [e^{A_{i_L} h_{i_L}} ... e^{A_{i_1} h_{i_1}} W_{h_0}^{i_0},...,W_{h_1}^{i_1}]a \tag{1}$$

where $a = [a_1^T,...,a_{L+1}^T]^T$ and $W_t^k = \int_0^t e^{A_k (t-\tau)} B_k B_k^T u(\tau) e^{A_k^T (t-\tau)} d\tau$. It is clear that if system (1) has feasible solution(s) then the designed switching signal and control strategy successfully realize the given controllability problem.

3 Proposed Approach

3.1 Constrained Genetic Algorithms

GA is one of the most efficient derivative-free and global optimizers that has been successfully applied in many applications. In an ordinary use of GA, the problem is unconstrained and only the decision variables (genes) are bounded. For this case, several modified versions of GA have been proposed. But if the optimization problem is constrained, the ordinary crossover and mutation operators can not be used and handling these constraints may be difficult. Generally constrained GA is a challenging problem and it is still open for further investigations. As mentioned in [11-13], there are two major approaches for handling constraints in GA: A. Direct constraint handling which contains four methods: 1.eliminating infeasible candidates, 2. repairing infeasible candidates, 3. preserving feasibility by special operators, 4. decoding such as transforming the search space. Direct constraint handling has two advantages. It might perform very well, and might naturally accommodate existing heuristics. Regarding the technique of direct constraint handling is usually problem dependent, designing a method for a given problem may be difficult and using a given method might be computationally expensive. B. Indirect constraint handling: Indirect constraint handling incorporates constraints into a fitness function. Advantages of indirect constraint handling are: generality (problem independent penalty functions), reducing problem to 'simple' optimization, and allowing

user preferences by selecting suitable weights. One of disadvantages of indirect constraint handling is loss of information by packing everything in a single number and generally in case of constrained optimization, it is reported to be weak. Generally, if suitable operators can be found for direct handling, undoubtedly direct search in the feasible space is much more efficient and faster than indirect type.

The genetic algorithms toolbox of the recent versions of MATLAB software (2007 and later) efficiently handles linear and nonlinear constraints. We recommend readers to use this toolbox especially for linear constraints since the toolbox handles linear constraints based on direct constraint handling and as a result, the population is kept feasible through generations. As direct constraint handling is faster and more effective than indirect type, we applied this toolbox in the simulation example at part 4.

Table 1 The commonest constraints of two levels

Constraints on switching signal (Top level)	
Dwell time	$h_k \geq dwell, \quad k = 0:L$
Maximum time to reach final state	$\sum_{k=0}^{L} h_k \leq T_{\max}$
Constraint on the order of index sequence	*For example subsystem 2 can be selected only after subsystem 1, etc.*
Constraints on control input (Bottom level)	
Constraint on vector a	$x_f - e^{A_{i_L} h_{i_L}} \ldots e^{A_{i_0} h_{i_0}} x_0 = [e^{A_{i_L} h_{i_L}} \ldots e^{A_{i_1} h_{i_1}} W_{h_0}^{i_0}, \ldots, W_{h_1}^{i_1}]a$
Constraint on control input amplitude	$u_{\min} \leq u = B_{i_k}^T e^{A_{i_k}^T (t_{k+1}-t)} a_{k+1} \leq u_{\max}$

3.2 Problem Formulation

According section 2, the decision variables as well as constraints can be categorized in two classes including switching signal design as the top level and control input design as the bottom level. Several constraints can be considered for switching signal design including desired dwell time, allowed maximum time to reach the final state, order of switching and etc. The switching indices are discrete variables when the switching durations are continuous. Totally in the top level, $2\times(L+1)$ decision variables are available including switching indices i_k and switching durations h_k. At the bottom level, as equation (1) shows, a is an $n\times(L+1)$ dimensional vector of decision variables with a linear equality constraint. Note that if u is constrained (such as actuator saturation), this can be simply handled in design process as a linear constraint on a (Table 1). The objective function can be generally formulated as a nonlinear (analytical or non-analytical) function of decision variables. In this paper, although we just consider single objective control design, note that the proposed approach is flexible enough to be applied for multi-objective design. Table 1 shows the details in mathematical form. Note using a simple rounding,

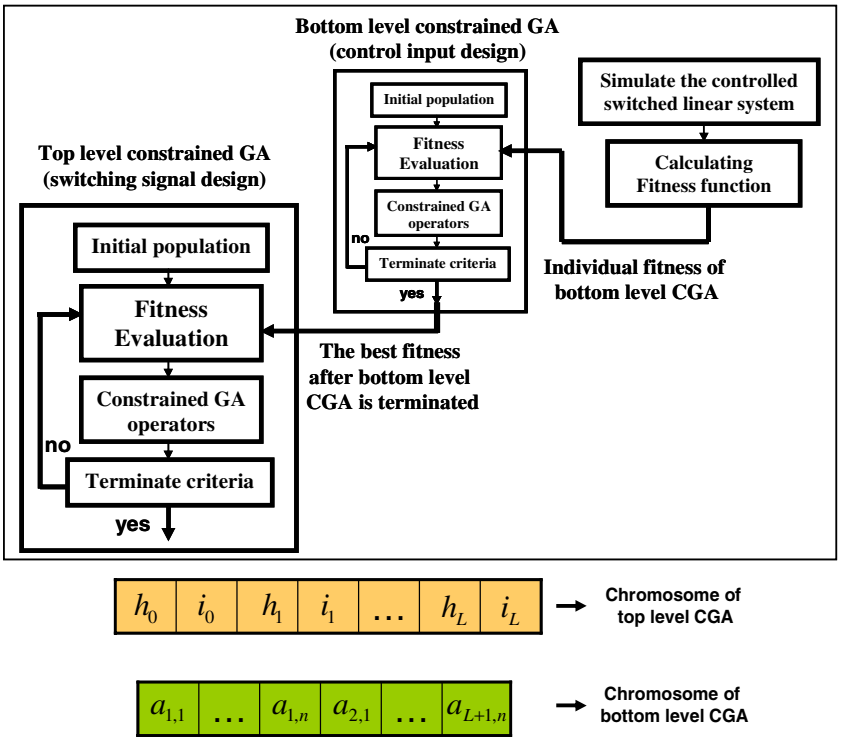


Fig. 2 Proposed MLCGA (top), Chromosomes architecture (bottom)

switching indices can be supposed continuous through optimization and converted to integer just for fitness evaluation.

3.3 Path Planning for Controllability Using MLCGA

Regarding the problem formulation in the previous subsection, this problem must be solved in two levels. From optimization perspective, the problem of each level is constrained and may be nonlinear and multi-modal as well as non-analytical. In these reasons, we propose multi-level constrained genetic algorithms to solve this problem. The top level CGA searches in the feasible space of switching signals. To evaluate the fitness of any chromosome of top level CGA, optimal control input must be found for this candidate switching signal. It means that for any fitness evaluation of top level CGA, a bottom level CGA must be completely run and terminate. Although several termination criterions are available, for the bottom level CGA, we recommend using pre-determined maximum number of generations. The main reason of this recommendation is ensuring a limited time for termination. If you apply another termination criterion, bottom level CGA may be terminated after a long time or even it may never be terminated. Figure 2 shows the MLCGA flowchart as well as chromosomes architecture. Depending on the

objective function, if the problem of control input design is analytical, algebraic and single modal, one can replace the bottom level CGA with a conventional nonlinear programming method. Although this recommendation can be performed for specific problems, it can effectively increase the speed of convergence.

4 Simulation Example

In this part, we design an optimal control system including optimal supervisor design (switching signal) as well as optimal controller design (control input) for the 3rd order switched linear system given by

$$A_1 = 0, B_1 = e_1, A_j = e_j e_{j-1}^T, B_j = 0,\ j = 2,3 \quad x_0 = \begin{bmatrix} 4 \\ 5 \\ 4 \end{bmatrix} \quad x_f = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{2}$$

where e_j is the unit column vector with the jth entry equal to one. According to Table 1, for this example, $L = 11,\ M = n = 3,\ T_{\max} = 30,\ dwell = 0.5.$ The order of switched subsystems is restricted to {1, 2, 3, 1, 2, 3 …}. There is no constraint on amplitude of control input. Maximum number of generations and population size are 50 and 10 for bottom level CGA and 30 and 20 for top level CGA. The objective is minimizing the value of the integral of 2-norm of state signal. We used Matlab 2007a (GA toolbox as well as simulink) for this example as represented in Figure 3. The following figures show the results. As the figures indicate, the final

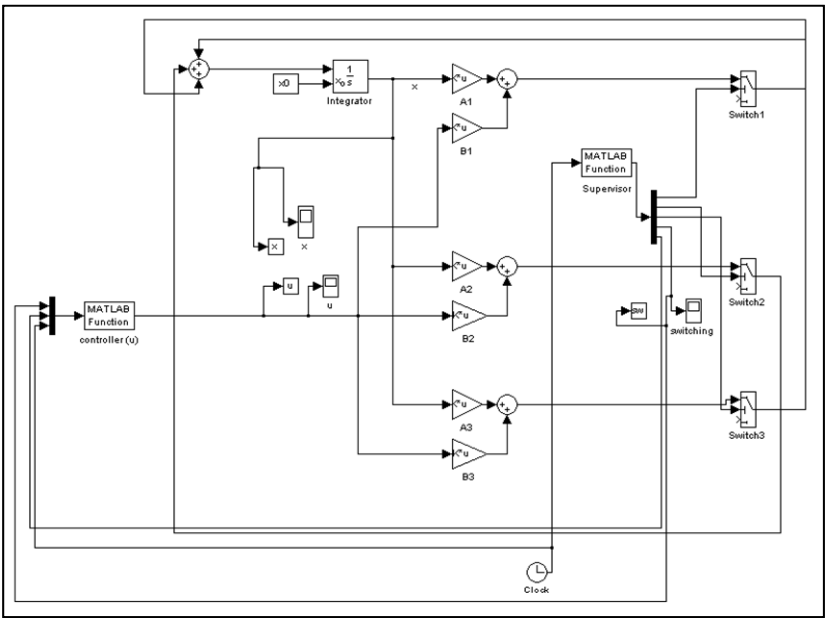


Fig. 3 Simulink file which is used for simulation example (Matlab 2007a)

time is very small and less than 8 seconds. The switching signal has a dwell time larger than 0.5. The 3d plot clearly shows the optimally designed path which achieved using the proposed method.

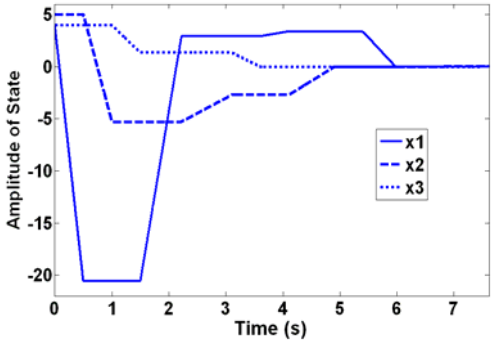


Fig. 4 State signals reach the final state in less than 8 seconds

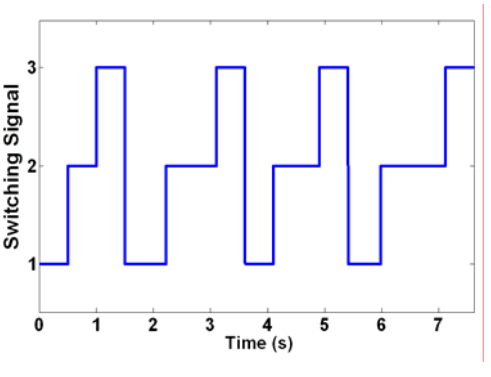


Fig. 5 Switching signal with dwell time larger than 0.5

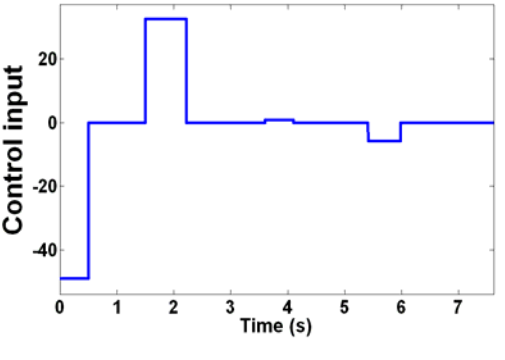


Fig. 6 Control input signal

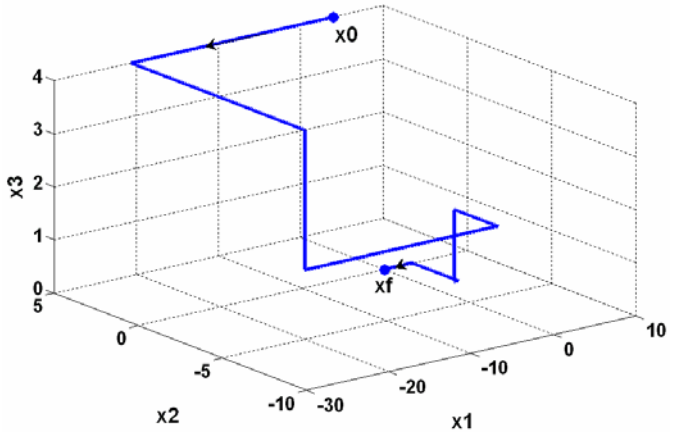


Fig. 7 Optimal path which was designed in the example for controllability of the switched linear system represented in equation (2)

5 Conclusion

In this paper, an optimal path including switching signal and control input is planned for controllability of a general switched linear system based on multi-level constrained genetic algorithms (MLCGA). The problem is solved in two levels. Switching signal design as top level and control input design as bottom level. The problem has several linear constraints such as controllability condition, control input amplitude (actuator saturation) and desired dwell time as well as desired final time. In this paper we formulate this problem as a general nonlinear constrained optimization problem that maybe multi-modal and non-analytical. As simulation example shows, using MLCGA an optimal path can be efficiently found with satisfying all constraints as well as desired performance. Generally the proposed approach can be successfully used for real world applications when the problem of path planning for controllability is complex.

References

1. Sun, Z., Ge, S.S.: Switched linear systems, control and design. Springer, London (2005)
2. Ge, S.S., Sun, Z., Lee, T.H.: Reachability and controllability of switched linear systems. In: Proceedings of the 2001 American Control Conference, vol. 3, pp. 1898–1903
3. Hihi, H., Rahmani, A.: A Sufficient and necessary conditions for the controllability of switching linear systems. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 3984–3989 (2007)
4. Ji, Z., Wang, L., Guo, X.: On Controllability of Switched Linear Systems. IEEE Transactions on Automatic Control 53(3), 796–801 (2008)

5. Hu, Z., Zhang, B., Deng, W.-h.: Output controllability of switched power converters as switched linear systems. In: The 4th International Power Electronics and Motion Control Conference, vol. 3, pp. 1665–1668 (2004)
6. Xie, G., Wang, L.: Controllability of periodically switched linear systems with saturating actuators. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 1660–1665 (2004)
7. Liang, L.v., Lin, Z., Brown, C.L.: Design of Switched Linear Systems in the Presence of Actuator Saturation. In: IEEE International Conference on Control and Automation, ICCA 2007, pp. 2659–2663 (2007)
8. Li, J., Han, X., Li, Y., Gao, L.: Optimal switched law of switched linear systems based on convergence route. In: Fifth World Congress on Intelligent Control and Automation, vol. 1, pp. 48–52 (2004)
9. Chen, D., Chen, Y.: Switched system optimal control based on dynamical programming principle and evolutional computation. In: Fifth World Congress on Intelligent Control and Automation, vol. 2, pp. 1104–1108 (2004)
10. Ji, Z.: Number of Switchings and Design of Switching Sequences for Controllability of Switched Linear Systems. In: Control Conference, CCC 2006, Chinese, pp. 1049–1054 (2006)
11. Craenen, B.G.W., Eiben, A.E., van Hemert, J.I.: Comparing Evolutionary Algorithms on Binary Constraint Satisfaction Problems
12. Santos, A., Dourad, A.: Constrained GA applied to Production and Energy Management of a Pulp and Paper Mill
13. Schoenauer, M., Xanthakis, S.: Constrained GA optimization. In: Proceedings of the 5th International Conference on Genetic Algorithms, Urbana Champaign (1993)

Particle Swarm Optimization for Inference Procedures in the Generalized Gamma Family Based on Censored Data

Mauro Campos, Renato A. Krohling*, and Patrick Borges

Abstract. The generalized gamma distribution offers a highly flexible family of models for lifetime data and includes a considerable number of distributions as special cases. This work deals with the use of the particle swarm optimization (PSO) algorithm in the maximum likelihood estimation of distributions of the generalized gamma family (*GG*-family) based on data with censored observations. We also discuss a procedure for testing whether a distribution that belong to *GG*-family is appropriate for lifetime data using the generalized likelihood ratio test principle. Finally, we present two illustrative applications using real data sets. For each data set, we use the PSO algorithm to fit several distributions of the *GG*-family simultaneously. Then, we test the appropriateness of each fitted model and select the most appropriate one using the Bayesian information criterion or the Akaike information criterion.

1 Introduction

The generalized gamma distribution (*GG*-distribution) was introduced by Stacy [1]. It offers a highly flexible family of models for lifetime data and also includes a considerable number of distributions as special cases [2]. Since the first published paper by Stacy in 1962, some attention has been given for the generalized gamma family (*GG*-family) and its applications in the literature. Stacy and Mihram [3] and Hager and Bain [4] present results related to the estimation of the parameters of the *GG*-distribution and discuss difficulties with convergence of algorithms for

Mauro Campos · Patrick Borges
Department of Statistics. Federal University of Espírito Santo. Av. Fernando Ferrari 514, Goiabeiras, CEP 29075-910, Vitória ES, Brazil
e-mail: maurocmc@cce.ufes.br, patrick@cce.ufes.br

Renato A. Krohling
Department of Informatics. Federal University of Espírito Santo. Av. Fernando Ferrari 514, Goiabeiras, CEP 29075-910, Vitória ES, Brazil
Tel.: +55 27 4009 2189
e-mail: krohling.renato@gmail.com

* Corresponding author.

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 411–422.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

maximum likelihood estimation (MLE). Prentice [5] and Lawless [6, 7] consider the *GG*-distribution in a reparameterized form to avoid difficulties in the MLE procedure. Di Ciccio [8] suggests an approximate method for inference within the same approach. Pham and Almhana [9] present a review of the basic properties related to the *GG*-distribution and prove some results on its hazard rate and stress-strength model. Dadpay et al. [10] study information properties of the *GG*-distribution and provide an assortment of information measures for the *GG*-family. Yacoub [11] introduces the α-μ distribution, which is in fact a rewritten form of the *GG*-distribution put in terms of a physical fading model.

MLE of the parameters of a *GG*-distribution based on censored data is a non-trivial optimization problem in which the point of absolute maximum of the log-likelihood function cannot be obtained explicitly by analytical methods. We need of iterative procedures to solve this problem. Gradient methods, such as the Newton-Raphson method and the method of scoring, require finding the score vector and the Hessian matrix associated to the log-likelihood function. However, these operations are hard to be realized analytically. Furthermore, gradient methods may fail to converge if the Hessian matrix is not positive definite [12, 13]. Particle swarm optimization (PSO) [14, 15] is a powerfull optimization method used in different areas of science and engineering. It is a direct-search method which does not require the evaluation of any partial derivatives of the objective function. This work deals with the use of the PSO algorithm in the MLE of models of the *GG*-family based on data with censored observations. As far as we know, for the first time introduced in the context of reliability engineering and lifetime data analysis. We also discuss a procedure for testing whether a distribution that belong to *GG*-family is appropriate for lifetime data by comparing it with other distributions in the same family. Formally, this procedure can be done using the generalized likelihood ratio test principle [13, Ch. 4]. In practice, we can use the PSO algorithm to fit several distributions simultaneously and then test the appropriateness of each fitted model. The most appropriate model can be selected using information criterions such as the Akaike criterion (AIC) [16] or the Schwarz criterion (BIC) [17].

The organization of the paper is as follows. In Section 2 the formulation of the problem is presented. Section 3 presents the *GG*-family and some of its properties. Section 4 discusses the MLE procedure, test for appropriateness of a family of distributions and selection of model. Section 5 introduces the PSO algorithm. Section 6 contains two illustrative applications using the *GG*-family to model lifetime data with censored observations and section 7 concludes the work.

2 Problem Formulation

Let T be a nonnegative random variable representing the lifetimes of elements in a population. Suppose that T is a continuous random variable with probability density function $f(\cdot;\theta)$ where $\theta = (\theta_1, \ldots, \theta_k)$ is a vector of parameters taking on values in a set Θ. Suppose further that t is a nonnegative real number. The probability that an element of the population survives longer than t is given by the reliability function

$$S(t;\theta) = Pr(T > t) = \int_t^{\infty} f(u;\theta)du. \tag{1}$$

Note that $S(0;\theta) = 1$ and $S(t;\theta) \to 0$ as $t \to \infty$. Another useful concept having to do with lifetime distributions is the hazard function defined by $h(t;\theta) = f(t;\theta)/S(t;\theta)$. The hazard function gives the risk of failure per unit time during the aging process. The functions f, S and h give mathematically equivalent specifications of the distribution of T. Given any one of them, the other two can be derived. The relationship among the three functions is estabilished by

$$f(t;\theta) = -\frac{dS(t;\theta)}{dt} = h(t;\theta)\exp\left(-\int_0^t h(u;\theta)du\right). \tag{2}$$

In statistical inference we use a sample of data to draw inferences about some aspect of the distribution from which the data were taken. Often the inference concerns the value of one or more unknown parameters, which describe some atribute of the distribution. However, lifetime data come with a feature that creates special problems in the analysis of the data. This feature is known as censoring and, broadly speaking, occurs when exact lifetimes are known for only a portion of elements under study. Formally, an observation is said to be right censored at C if the exact value of the observation is not known but only that it is greater than or equal to C. Similarly, an observation is said to be left censored at C if it is known only that the observation is less than or equal C. For convenience only right censoring will be discussed here, though many of ideas transfer in an obvious way to the case of left censoring. In addition, the term *censoring* will be used, meaning in all situations *right censoring*, and when an element has his lifetime censored at C, we will call C the censoring time for the element. To discuss censoring we must consider the way in which data are obtained. In fact, censoring arises for a variety of reasons, and we consequently distinguish among several types of censoring processes in the discussion that follows. A type I censored sample is one that arises when n elements under study in a life test experiment are subjected to limited periods of observation $C_1,\ldots,C_n$, so that the lifetime T_i of an element is observed only if $T_i \leq C_i$. Otherwise, the lifetime for the ith element is censored at C_i. When $C_i = C$ for all i, we say that the data are singly type I censored. It should be noted that with type I censoring the number of exact lifetimes observed is random. A type II censored sample is one that arises when n elements are placed on test, but instead of continuing until all n elements have failed, the test is terminated at the time of the rth element failure. Observe that with type II censoring the number of exact lifetimes is fixed, in contrast to the case of type I censoring, where it is random. A type III censored sample is one in which each element under study in a life test experiment is assumed to have a lifetime T and a censoring time C, where T and C are independent continuous random variables. The notation $(t,\delta) = \{(t_i,\delta_i); i = 1,\ldots,n\}$ will be used to represent a set of n observations from the distribution of T, where $\delta_i = 0$ if t_i is a censored observation and $\delta_i = 1$ otherwise. In the context of censored data, the likelihood function is defined by

$$L(\theta) = L(\theta;t) \propto \prod_{i=1}^{n} [f(t_i;\theta)]^{\delta_i} [S(t_i;\theta)]^{1-\delta_i} \tag{3}$$

and the log-likelihood function is defined by $l(\theta) = \log L(\theta)$.

This work deals with the use of the PSO algorithm to find the value of θ that maximizes $l(\theta)$ (or $L(\theta)$) when $f(\cdot;\theta)$ is the probability density function of the *GG*-distribution or some other distribution that belong to *GG*-family. We also analyse two real data sets and demonstrate how the methodology can be used in practice. For each lifetime data set, we use the PSO algorithm to fit several distributions of the *GG*-family simultaneously. Then, we test the appropriateness of each fitted model and select the most appropriate one based on information criterions such as AIC or BIC.

3 Generalized Gamma Family

The probability density function of the *GG*-distribution is given by

$$f(t) = \frac{\gamma}{\beta^{\gamma\eta}\Gamma(\eta)} t^{\gamma\eta-1} \exp\left[-\left(\frac{t}{\beta}\right)^{\gamma}\right], \qquad t \geq 0, \tag{4}$$

where $\gamma, \eta, \beta > 0$ and $\Gamma(\eta)$ is the well-known gamma function. We note that γ and η are shape parameters, and β is the scale parameter. The notation $T \sim GG(\gamma, \eta, \beta)$ will be used to indicate that a random variable T has a *GG*-distribution with parameters γ, η and β. The *GG*-family includes several models as special cases. Some subfamilies considered in statistics and other related fields are: exponential ($\gamma = \eta = 1$), Weibull ($\eta = 1$) and gamma ($\gamma = 1$), half-normal ($\gamma = 2$ and $\eta = 1/2$), Rayleigh ($\gamma = 2$ and $\eta = 1$), Maxwell-Boltzmann ($\gamma = 2$ and $\eta = 3/2$) and chi-squared ($\gamma = 1$, $\beta = 2$ and $\eta = k/2, k = 1, 2, \ldots$). In addition, the lognormal distribution appears as a limiting case when $\eta \to \infty$ [7]. Now, we present some useful results related to *GG*-family. If $T \sim GG(\gamma, \eta, \beta)$, then $T^s \sim GG(\gamma/s, \eta, \beta^s)$ for all $s > 0$. In particular, $T^\gamma \sim G(\eta, \beta^\gamma)$, where $G(\eta, \beta^\gamma)$ denotes the gamma distribution with shape parameter η and scale parameter β^γ. The reliability function of the *GG*-distribution is given by

$$S(t;\gamma,\eta,\beta) = 1 - I\left(\eta, \frac{t^\gamma}{\beta^\gamma}\right) \tag{5}$$

where $I(\eta,\tau) = \int_0^\tau (1/\Gamma(\eta)) u^{\eta-1} e^{-u} du$ is the incomplete gamma function. It is important to note that $I(\eta,\tau) = Pr(G_\eta \leq \tau)$, where $G_\eta \sim GG(1,\eta,1)$. Finally, the log-likelihood function of the *GG*-distribution is given by

$$\begin{aligned} l_{GG}(\gamma,\eta,\beta) = {} & (\log\gamma - \log\Gamma(\eta) - \gamma\eta\log\beta) \sum_{i=1}^{n} \delta_i + (\gamma\eta - 1) \sum_{i=1}^{n} \delta_i \log t_i \\ & - \beta^{-\gamma} \sum_{i=1}^{n} \delta_i t_i^\gamma + \sum_{i=1}^{n} (1-\delta_i) \log S(t_i;\gamma,\eta,\beta). \end{aligned} \tag{6}$$

4 Inference Procedures

The maximum likelihood estimate of θ, denoted by $\hat{\theta}_n$, is the value of θ that maximizes the log-likelihood function $l(\theta)$. Given the usual regularity conditions[1], it follows that the maximum likelihood estimator is asymptotically normally distributed [13, Ch. 3]. We use the notation $\hat{\theta}_n \overset{a}{\sim} N_k(\theta, \Sigma_\theta)$ to indicate this result. The symbol $N_k(\theta, \Sigma_\theta)$ represents a multivariate normal distribution with mean θ and variance-covariance matrix Σ_θ, where Σ_θ is the inverse of the (Fisher) information matrix. As a consequence, we can construct an asymptotic confidence interval for $g(\theta)$ where $g : \mathbb{R}^k \to \mathbb{R}$ is a smooth function such that $\nabla g(\hat{\theta}_n) \neq 0$. In fact, if $z_{\alpha/2}$ is the $(1-\alpha/2)$th quantile of the standard normal distribution, $\hat{\sigma}_g = \hat{\sigma}(g(\hat{\theta}_n)) = \sqrt{(\nabla g(\hat{\theta}_n))^T \Sigma_{\hat{\theta}} \nabla g(\hat{\theta}_n)}$ and

$$J = (g(\hat{\theta}_n) - z_{\alpha/2} \cdot \hat{\sigma}_g; g(\hat{\theta}_n) + z_{\alpha/2} \cdot \hat{\sigma}_g), \tag{7}$$

then $Pr(J \text{ contains } g(\theta)) \to 1-\alpha$ as $n \to \infty$.

A procedure for testing whether a distribution is appropriate for observed data is to compare the distribution with a more general family that includes the distribution of interest as a special case. As pointed out in section 3, the *GG*-family is a rich family and includes a considerable number of distributions as special cases. Thus, we can use this fact to evaluate the appropriateness of a distribution that belong to *GG*-family for lifetime data. Formally, let $\mathscr{F}_\theta$ be a family of distributions parameterized by a finite number of parameters, $\theta_1, \ldots, \theta_k$. Suppose that f' is a subfamily of $\mathscr{F}_\theta$. We wish to test whether the observed data are from f'. Formally, we wish to test

$$H_0 : \theta_{i_1} = \theta_{i_1}^0, \ldots, \theta_{i_q} = \theta_{i_q}^0, \theta_{i_{q+1}}, \ldots, \theta_{i_k}, \tag{8}$$

where $\theta^0 = (\theta_{i_1}^0, \ldots, \theta_{i_q}^0)$ is a vector of specific numbers, $\theta^* = (\theta_{i_{q+1}}, \ldots, \theta_{i_k})$ is left unspecified and $q < k$. The generalized likelihood ratio test [13, Ch. 4] can be used in this case. The test statistic is given by

$$\Lambda = -2\log\left(\frac{L(\hat{\theta}_n^*(\theta_0), \theta_0)}{L(\hat{\theta}_n)}\right) = 2\left[l(\hat{\theta}_n) - l(\hat{\theta}_n^*(\theta_0), \theta_0)\right] \tag{9}$$

where $\hat{\theta}_n$ is the maximum likelihood estimator of θ and $\hat{\theta}_n^*(\theta_0)$ is the maximum likelihood estimator of θ^* given θ_0. Under H_0, Λ have an asymptotic chi-squared distribution with degrees of freedom equal q [13, Ch. 4]. Thus, for a given significance level α, H_0 is rejected if $\Lambda > \chi^2_{(q),\alpha}$, where $\chi^2_{(q),\alpha}$ is the $(1-\alpha)$th quantile of chi-squared distribution with q degrees of freedom.

It is important to note that failure to reject H_0 does not imply that the distribution under the null hypothesis provides a perfect fit to the data. On the other hand, rejection of H_0 does not mean that the distribution under the alternative hypothesis is the best choice either. In practice, we fit several distributions simultaneously and

[1] These conditions are essentially smoothness conditions on $f(\cdot, \theta)$.

then test the appropriateness of each fitted model. The most appropriate model can be selected using information criterions. Akaike [16] indicates that, given a class of competing models for a data set, one choose the model that minimizes

$$\text{AIC} = -2l(\hat{\theta}_n) + 2(\text{number of parameters}). \tag{10}$$

Another method is the Schwarz criterion [17]. In the Schwarz's approach one choose the model that minimizes

$$\text{BIC} = -2l(\hat{\theta}_n) + (\log n)(\text{number of parameters}). \tag{11}$$

5 Particle Swarm Optimization (PSO)

Let $L(\theta)$ be a real-valued function defined on a set $\Theta \subseteq \mathbb{R}^k$. A point θ^* in Θ is said to be a point of local maximum of L if there exists a neighborhood $N_\delta(\theta^*) = \{\theta \in \mathbb{R}^k; \|\theta - \theta^*\| < \delta\} \subseteq \Theta$ such that $L(\theta) \leq L(\theta^*)$ for all $\theta \in N_\delta(\theta^*)$. If $L(\theta) \geq L(\theta^*)$ for all $\theta \in N_\delta(\theta^*)$, then θ^* is a point of local minimum. If one of theses inequalities holds for all θ in Θ, then θ^* is called a point of absolute maximum, or a point of absolute minimum, respectively, of L in Θ. In either case, θ^* is referred to as a point of optimum, and the value of $L(\theta)$ at $\theta = \theta^*$ is called an optimum value of $L(\theta)$.

PSO [14, 15] is a powerfull optimization method used in different areas of science and engineering in situations in which the optimum can not be obtained explicitly by analytical methods. PSO is a population-based algorithm. It is initialized with a population of candidate solutions. Each candidate solution in PSO, called *particle*, has associated a randomized velocity. Each particle moves through the search space and keeps track of its coordinates in the search space, which is associated with the best solution (fitness) it has achieved so far, *pbest*. Another *best* value tracked by the *global* version of the particle swarm optimizer is the overall best value, *gbest*, and its location, obtained so far by any particle in the population. The algorithm proceeds updating first all the velocities and then the positions of the particle as follows:

$$\omega_i^{t+1} = K\left[\omega_i^t + c_1 U_1(p_i - \theta_i^t) + c_2 U_2(p_g - \theta_i^t)\right] \tag{12}$$

$$\theta_i^{t+1} = \theta_i^t + \omega_i^{t+1}, \tag{13}$$

with $K = 2/|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|$ and $\varphi = c_1 + c_2 > 4$, where K is a function of c_1 and c_2 [18].

The vector[2] $\theta_i = (\theta_{i1}, \ldots, \theta_{ik})$ stands for the position of the ith particle, $\omega_i = (\omega_{i1}, \ldots, \omega_{ik})$ stands for the velocity of the ith particle and $p_i = (p_{i1}, \ldots, p_{ik})$ represents the best previous position (the position giving the best fitness value) of the ith

[2] We use the notation θ and ω for position and velocity respectively instead of x and v, as usual in the PSO literature, because the variable to be optimized is θ.

particle. The index g represents the index of the best particle among all the particles in the swarm. The constants c_1 and c_2 are positive numbers; U_1 and U_2 are random numbers in the range $[0,1]$ generated according to a uniform probability distribution.

The *cognition* term $c_1U_1(p_i - \theta_i^t)$ represents the independent behaviour of the particle itself. The *social* term $c_2U_2(p_g - \theta_i^t)$ represents the collaboration among the particles. The constants c_1 and c_2 represent the weighting of the *cognition* and *social* parts that pull each particle toward *pbest* and *gbest* positions.

A detailed discussion of the constriction factor K, introduced by Clerc and Kennedy [18], is beyond the scope of this paper. For more details, the reader is referred to [18]. Usually, when the constriction factor is used, φ is set to 4.1 ($c_1 = c_2 = 2.05$), and the constriction factor K is 0.729. The PSO algorithm is described as follows.

Algorithm

- **Input parameters:** Swarm size, K, c_1, c_2
- **For** each particle i
- // random initialization of a population of particles with positions θ_i using uniform probability distribution
- $\theta_i = \underline{\theta}_i + (\overline{\theta}_i - \underline{\theta}_i)U_i(0,1)$
- $p_i = \theta_i$
- Compute $L(\theta_i)$ // fitness evaluation
- $p_g = \arg\max L(\theta_i)$ // global best particle
- **End for**
- **Do**
- **For** each particle i
- Update velocity and position according to (12) and (13) respectively
- Compute $L(\theta_i)$ // fitness evaluation
  - **If** $L(\theta_i) > L(p_i)$ **then** $p_i = \theta_i$ // update of the personal best
  - **If** $L(\theta_i) > L(p_g)$ **then** $p_g = p_i$ // update of the global best
- **End for**
- **While** termination condition not met.
- **Output:** $p_g = \theta^*$.

The termination criterion of the PSO algorithm is usually the maximum number of the iteration, which is pre-specified by the designer.

6 Applications

In this section, we present two illustrative applications using real data sets. For each data set, we use the PSO algorithm to obtain the maximum likelihood estimates and the log-likelihoods for the exponential, Weibull, gamma, lognormal and generalized

gamma distributions. Furthermore, we test the appropriateness of each fitted model and select the most appropriate one based on information criterions.

Example 1. Consider the data set in [19]. These type I censored data consist of times between failures for 10 machining centers subjected to a lifetime test over half a year. The times are given in Table 1. Observations with the symbol "+" represent censored times.

Table 1 Times between failures of machining center

176.00	248.00	10.50	472.00	45.00	39.00	209.33	261.25	510.00	120.00	224.00
267.50	32,00	50.00	138.50	398.00	478.00	353.00	348.00	137.06	332.50+	84.00+
267.00+	165.00+	283.16+	700.00+	562.00+	387.00+	383.00+	1.50+			

Table 2 Lifetimes of 34 transistors in an accelerated life test

03	04	05	06	06	07	08	08	09	09	09	10	10	11	11	11	13	13	13	13	13	17	17	19	19	25	29	33	42	42	52
52+	52+	52+																												

Example 2. Consider the data set in [7, page 208] (or originally in [20]). These type II censored data consist of lifetimes of 34 transistors in an accelerated life test. The times, in weeks, are given in Table 2. Once again, times with the symbol "+" represent censored observations.

Table 3 Maximum likelihood estimates - Ex. 1

Model	$\hat{\gamma}$	$\hat{\eta}$	$\hat{\beta}$	log-L
Exp	-	-	384.11	-139.02
Wei	1.20	-	377.00	-138.59
G	-	1.42	240.11	-138.82
GG	2.11	0.48	601.00	-138.40
LN	$\hat{\mu} = 5.54$, $\hat{\sigma} = 1.24$			-140.03

log-L, log-likelihood.

Table 4 Maximum likelihood estimates - Ex. 2

Model	$\hat{\gamma}$	$\hat{\eta}$	$\hat{\beta}$	log-L
Exp	-	-	20.74	-124.99
Wei	1.22	-	21.70	-124.04
G	-	1.62	12.40	-123.11
GG	0.37	10.09	0.03	-120.98
LN	$\hat{\mu} = 2.68$, $\hat{\sigma} = 0.83$			-119.90

log-L, log-likelihood.

For each data set, we use the PSO algorithm to obtain the maximum likelihood estimates and the log-likelihoods for the exponential (Exp(β)), Weibull (Wei(γ,β)), gamma (G(η,β)), lognormal (LN(μ,σ)) and generalized gamma (GG(γ,η,β))

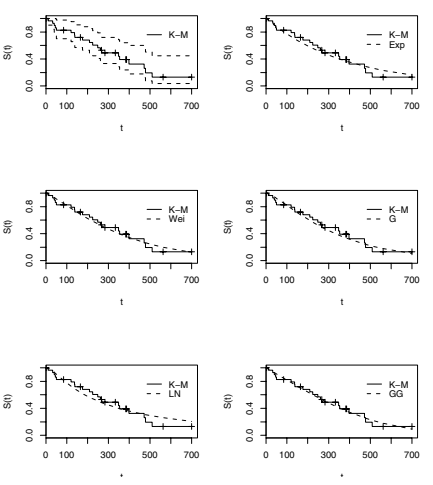


Fig. 1 Kaplan-Meier estimate of the reliability function and fitted reliability functions (Ex. 1)

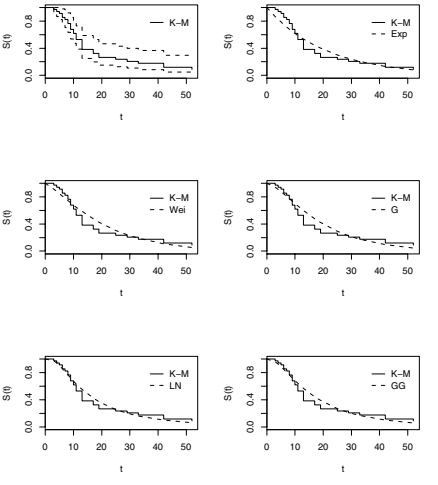


Fig. 2 Kaplan-Meier estimate of the reliability function and fitted reliability functions (Ex. 2)

distributions. The parameters of the PSO algorithm used are the same as described in section 5. The number of particles used was set to 30, and the maximum number of iteration set to 100. Typically the PSO algorithm converges in the first 50 iterations. The results obtained are given in Tables 3 and 4. Fig. 1 shows the Kaplan-Meier (K-M) estimate [21, 7] of the reliability function and the fitted reliability function of each model mentioned, considering the data set of the example 1. Fig. 2 shows the analogous results for the data set of the example 2. The results of goodness of fit tests based on asymptotic likelihood inferences are given in Tables 5 and 6 jointly with the BIC scores and the AIC scores.

Consider a significance level of the 5% for the example 1. From the Table 5, we see that all hypothesized distributions can be considered appropriate for the data of the example 1. Because of its simplicity (principle of parsimony), we may select the exponential distribution as our choice of model for this data set. On the other hand, the exponential distribution would be selected by either the BIC or AIC procedure, which is consistent with the results obtained in the goodness of fit tests. An approximate confidence interval for the parameter β of the exponential distribution can be obtained using the normal approximation to the distribution of $\hat{\beta}$. The observed (Fisher) information can be found so that the variance of $\hat{\beta}$ is given by $var(\hat{\beta}) = \hat{\beta}^2/20$. Then, an approximate 95% confidence interval for β is given by $\hat{\beta} \pm 1.96\sqrt{var(\hat{\beta})} = (215.77; 552.45)$.

Consider a significance level of the 5% for the example 2. From the Table 6, we see that the exponential distribution is not rejected in favor of the Weibull distribution ($\Lambda = 1.919$, p-value $= 0.166$) or in favor of the gamma distribution ($\Lambda = 3.783$, p-value $= 0.052$). The hypothesis that the underlying distribution is

Table 5 Goodness-of-fit tests and model selection - Ex. 1

Model	Λ	p-value	BIC	AIC
Exp[a]	0.861	0.354	-	-
Exp[b]	0.399	0.528	-	-
Exp[c]	1.246	0.536	281.439	280.038
Wei	0.385	0.535	283.979	281.177
G	0.847	0.357	284.441	281.637
LN	3.271	0.071	286.865	284.062
GG	-	-	286.995	282.791

Λ, likelihood ratio test statistic.
p-value, $P(\chi^2 > \Lambda)$.
a Relative to the Weibull.
b Relative to the gamma.
c Relative to the generalized gamma.

Table 6 Goodness-of-fit tests and model selection - Ex. 2

Model	Λ	p-value	BIC	AIC
Exp[a]	1.919	0.166	-	-
Exp[b]	3.783	0.052	-	-
Exp[c]	8.033	0.018	253.520	251.994
Wei	6.114	0.013	255.127	252.074
G	4.250	0.039	253.264	250.211
LN	-2.157	1	246.857	243.804
GG	-	-	252.540	247.961

Λ, likelihood ratio test statistic.
p-value, $P(\chi^2 > \Lambda)$.
a Relative to the Weibull.
b Relative to the gamma.
c Relative to the generalized gamma.

the exponential versus the alternative hypotheses that the distribution is the generalized gamma is rejected ($\Lambda = 8.033$, p-value $= 0.018$). Furthermore, the Weibull and gamma distributions are also rejected in favor of the generalized gamma ($\Lambda = 6.114$, p-value $= 0.013$ and $\Lambda = 4.250$, p-value $= 0.039$ respectively). This implies that the exponential distribution may not be an appropriate distribution since the Weibull and gamma distributions are rejected. The lognormal distribution is not rejected in favor of the generalized gamma distribution. Therefore, we may accept the lognormal distribution as our choice of model for the data set. On the other hand, the lognormal distribution would be selected by either the BIC or AIC procedure, which is consistent with the results obtained in the goodness of fit tests. Interval estimation for the parameters μ and σ of the lognormal distribution can be obtained using the normal approximation to the distribution of $(\hat{\mu}, \hat{\sigma})$. The observed (Fisher) information matrix can be found so that

$$\Sigma_{\hat{\mu},\hat{\sigma}} = \begin{pmatrix} 0.022799 & 0.000392 \\ 0.000392 & 0.000947 \end{pmatrix}. \tag{14}$$

An approximate 95% confidence interval for the parameter μ is given by $\hat{\mu} \pm 1.96\sqrt{var(\hat{\mu})} = (2.38; 2.98)$ and an approximate 95% confidence interval for the parameter σ is given by $\hat{\sigma} \pm 1.96\sqrt{var(\hat{\sigma})} = (0.77; 0.89)$.

7 Conclusion

This work deals with the problem of the MLE of the *GG*-distribution and other distributions of the *GG*-family based on censored data. This is a nontrivial optimization problem in which the point of absolute maximum of the log-likelihood function cannot be obtained explicitly by analytical methods. As a consequence, we propose an approach for this problem based on evolutionary computation. Specifically, we use the PSO algorithm to maximize the likelihood functions of the *GG*-family based on censored data. The good performance of the PSO algorithm indicates that the method works quite well. We present two illustrative applications using distributions of the *GG*-family to model lifetime data. For each data set, we use the PSO algorithm to obtain the maximum likelihood estimates and the log-likelihoods for several models of the *GG*-family, including the *GG*-distribution as well. Furthermore, we test the appropriateness of each fitted model, using the generalized likelihood ratio test principle, and select the most appropriate one based on information criterions. This paper shows the potencial of the PSO algorithm to solving some problems in the field of the reliability engineering and lifetime data analysis. Work in progress is investigating the use of improved versions of PSO, e.g. [22, 23, 24] to tackle other problems in the field of statistics and reliability engineering.

References

1. Stacy, E.: A generalization of the gamma distribution. Annals of Mathematical Statistics 33, 1187–1192 (1962)
2. Johnson, N., Kotz, S., Balakrishnan, N.: Continuous Univariate Distributions, 2nd edn., vol. 1. John Wiley and Sons, New York (1994)
3. Stacy, E., Mihram, G.: Parameter estimation for a generalized gamma distribution. Technometrics 7, 349–358 (1965)
4. Hagar, H., Bain, L.: Inferential procedures for the generalized gamma distribution. Journal of the American Statistical Association 65, 1601–1609 (1970)
5. Prentice, R.: A log gamma model and its maximum likelihood estimation. Biometrika 61, 539–544 (1974)
6. Lawless, J.: Inference in the generalized gamma and log gamma distributions. Technometrics 22, 409–419 (1980)
7. Lawless, J.: Statistical Models and Methods for Lifetime Data. John Wiley and Sons, New York (1982)
8. Di Ciccio, T.: Approximate inference for the generalized gamma distribution. Technometrics, 33–40 (1987)
9. Phan, T., Almhana, J.: The generalized gamma distribution: its hazard rate and stress-strength model. IEEE Transactions on Reliability 44, 392–397 (1995)

10. Dadpay, A., Soofi, E., Soyer, R.: Information measures for generalized gamma family. Journal of Econometrics 56, 568–585 (2007)
11. Yacoub, M.: The α-μ distribution: a physical fading model for the Stacy distribution. IEEE Transactions on Vehicular Technology 56, 27–34 (2007)
12. Khuri, A.: Advanced Calculus with Applications, 2nd edn. John Wiley and Sons, New York (2003)
13. Garthwaite, P., Jolliffe, I., Jones, B.: Statistical Inference, 2nd edn. Oxford University Press, New York (2002)
14. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1941–1948. IEEE Service Center, Piscataway (1995)
15. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
16. Akaike, H.: Information theory and an extension of the maximum likelihood principle. In: Second International Symposium on Information Theory, pp. 267–281 (1973)
17. Schwarz, G.: Estimating the dimension of a model. Annals of Statistics 6, 461–464 (1978)
18. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a with multidimensional complex space. IEEE Transactions on Evolutionary Computation 6, 58–73 (2002)
19. Dai, Y., Zhou, Y.-F., Jia, Y.-Z.: Distribution of time between failures of machining center based on type I censored data. Reliability Engineering and System Safety 79, 377–379 (2003)
20. Wilk, M., Gnanadesikan, R., Huyett, M.: Estimation of parameters of the gamma distribution using order statistics. Biometrika 49, 525–545 (1962)
21. Kaplan, E., Meier, P.: Nonparametric estimation from incomplete observations. Journal of the American Statistical Association 53, 457–481 (1958)
22. Clerc, M.: Particle Swarm Optimization. ISTE Publishing Company, London (2006)
23. Kennedy, J., Mendes, R.: Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. IEEE Transactions on Systems, Man and Cybernetic 36, 515–519 (2006)
24. Krohling, R., Coelho, L.: Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems. IEEE Transactions on Systems, Man and Cybernetics 36, 1407–1416 (2006)

SUPER-SAPSO: A New SA-Based PSO Algorithm

Majid Bahrepour, Elham Mahdipour, Raman Cheloi, and Mahdi Yaghoobi

Abstract. Particle Swarm Optimisation (PSO) has been received increasing attention due to its simplicity and reasonable convergence speed surpassing genetic algorithm in some circumstances. In order to improve convergence speed or to augment the exploration area within the solution space to find a better optimum point, many modifications have been proposed. One of such modifications is to fuse PSO with other search strategies such as Simulated Annealing (SA) in order to make a new hybrid algorithm –so called SAPSO. To the best of the authors' knowledge, in the earlier studies in terms of SAPSO, the researchers either assigned an inertia factor or a global temperature to particles decreasing in the each iteration globally. In this study the authors proposed a local temperature, to be assigned to the each particle, and execute SAPSO with locally allocated temperature. The proposed model is called SUPER-SAPSO because it often surpasses the previous SAPSO model and standard PSO appropriately. Simulation results on different benchmark functions demonstrate superiority of the proposed model in terms of convergence speed as well as optimisation accuracy.

1 Introduction

Particle Swarm Optimisation and Simulated Annealing have their own advantages and drawbacks. PSO is related to the birds flocking; PSO seeks inside solution space to find the most optimistic result, however if it starts with inappropriate points, it's possible to get stuck into local optimums because of its high velocity; therefore the most problem with PSO is premature convergence [1, 2].

PSO cannot provide appropriate diversity while exploring the solution space because there is no diversity preservation operator to keep solutions in a diverse manner [3, 4]. Simulated Annealing (SA) is a kind of global optimisation technique based upon annealing of metal that uses random search to find optimal points [5, 6]. As might be expected, finding the global optimum in this way is not

Majid Bahrepour
Pervasive Systems Research Group, Twente University, The Netherlands
e-mail: m_bahrepour@ieee.org

Elham Mahdipour
Khavaran University, Mashhad, Iran

Raman Cheloi
Leiden University, the Netherlands

Mahdi Yaghoobi
Islamic Azad University, Mashhad, Iran

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 423 – 430.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

guaranteed, but there is usually appropriate diversity in searching solution space due to the hot temperature that lets particles move in any directions.

The SA-based particle swarm optimisation (SAPSO) fuses PSO with SA and often results in more optimised search than PSO and SA separately [2]. In this paper a novel version of SA-based PSO algorithm is proposed. Empirical results reveal that the proposed approach is highly optimised, as it often outperforms both the previous SAPSO model as well as the standard PSO approach.

This paper is structured as follows. In Section 2, standard PSO algorithm is described briefly. In Section 3, the previous SAPSO model is reviewed. In Section 4, the proposed SA-based algorithm, SUPER-SAPSO, is introduced. In Section 5, the experimental results are demonstrated and compared. Finally Section 6 discusses the results and presents some conclusions.

2 Particle Swarm Optimisation

According to Eberhart and Kennedy [1, 7, 8], PSO is a kind of evolutionary algorithms that discover the best solution by simulating the movement and flocking of birds [1]. Each particle is directed to special coordinate. These particles are moved toward the global optimum after some iteration. PSO optimisation, each particle has the ability to remember its previous best position (P_{Best}) and the global best position (G_{Best}). In addition, a new velocity value (V) is calculated based on its current velocity. The new velocity value is then used to compute the next position of particle in solution space. The original PSO's velocity formula is:

$$\begin{aligned} V[t+1] &= w.v[t] + c_1.rand(1).(G_{Best}[t] - \text{Present}[t]) \\ &\quad + c_2.rand(1).(P_{Best}[t] - \text{Present}[t]) \\ \text{Present}[t+1] &= \text{Present}[t] + V[t+1] \end{aligned} \tag{1}$$

Where $V[]$ is the velocity vector; variables c_1,c_2 are the acceleration constants and positive constants; *rand* is a random number between 0 and 1; Present[] is the position vector; W is the inertia weight.

W is not usually appear in standard PSO algorithm version. Searching inside the solution space using PSO, the exploration area is reduced gradually as the generation increasing which can be considered as a clustering strategy near the optimal point [1, 2].

3 SAPSO Hybrid Algorithm

SAPSO algorithm which is a combination of SA and PSO can avoid the key problem of PSO being premature convergence [2]. The premature convergence happens due to fast clustering of particles near the optimal point whilst the optimal point might be a local optimum. Therefore, PSO algorithm can potentially find the global optimum, but it is also possible to get stuck in local optimums. SA

algorithm can provide a better variety in seeking the solution space due to the hot temperature letting the particles move freely in any direction. Combination of PSO and SA can bear new benefits and compensate drawback of the both [2].

Similar to PSO algorithm, SAPSO searching process is started with random initialisation (dispersion) of particles. Firstly, each particle is moved by SA algorithm to a new position that augments variety of search which is accomplished by the use of Equation (2). Secondly, PSO will help the particles converge to global optimal by the use of Equation (1). This process is then repeated until a minimum error is achieved or maximum iterations are reached.

In the process of real annealing of metals, each new particle randomly is laid around the original particles. In this method variation range of original particles can be determined as a parameter like r_1, Expression (2) is a formulation for the variation of particles [2].

$$\text{Present}[t+1] = \text{Present}[t] + r_1 - r_1.2.rand(1) \tag{2}$$

Where parameter r_1 reduce gradually as the generation is increasing, *rand(*1*)* is random number between 0 and 1.

According to [2] the SAPSO algorithm is as follows:

1. Initialise *n* particles randomly.
2. Compute each particle's fitness.
3. Transform particles with the SA algorithm according to the Expression (2).
4. Compare particle's fitness evaluation with its personal best position (P_{Best}) if its fitness is better; replace P_{Best} with its fitness.
5. Compare particle's fitness evaluation with its global best position (G_{Best}) if its fitness is better; replace G_{Best} with its fitness.
6. Update each particle's velocity and position according to the Expressions (1).

This process continues until either the appropriate fitness is achieved or maximum iterations are reached.

4 Proposed Algorithm (SUPER-SAPSO)

Analogous with SAPSO algorithm, SUPER-SAPSO algorithm fuses SA with standard PSO, but particles' movement is done by the Expression (3) rather than Expression (2).

$$\text{Present}[t+1] = (\text{Present}[t] + v[t+1]).T \tag{3}$$

Where *Present[]* is the location vector, *v[]* is the velocity vector, and *T* is temperature of particle. Where *T* is a function of error and by the growth of error, *T* is increased as well.

SUPER-SAPSO algorithm is as follows:

1. Initialise a population of particles with random positions and velocities.

2. Calculate G_{Best} and assign T=1 to it.
3. For remaining particles evaluate fitness and assign temperature to them ($1 \leq T \leq 4$) so those particles with poorer fitness must have hotter temperature.
4. Transform particles according to the Expression (3).
5. For each particle, compares its fitness and its personal best position (P_{Best}) if current value is better than P $_{Best}$, the set P_{Best} equal to current value.
6. For each particle, compares its fitness and its global best position (G $_{Best}$) if current value is better than G_{Best}, the set G_{Best} equal to current value.
7. Update each particle's velocity and position according to the Expressions (1).
8. Best particle among n particles is recognised as leading particle to advance search process.
9. Go to Step 4, until the termination criterion is satisfied.

Termination criterion is either achievement of appropriate fitness or termination of computational time.

The main differences between SUPER-SAPSO algorithm and SAPSO algorithm are:

1. SUPER-SAPSO assigns the temperatures locally to the each particle.
2. The temperature is a function of error.

SUPER-SAPSO in comparison with SAPSO and standard PSO is investigated on benchmark functions and the experimental results are reported in the next section.

5 Experimental Results

Seven numeric optimisation problems were chosen to compare the relative performance of SUPER-SAPSO algorithm to SAPSO and PSO. These functions are standard benchmark test functions and are all minimisation problems.

The first test function is the generalised Rastrigin function:

$$F_1(x) = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i)) \tag{4}$$

The second function is the Foxholes function given by:

$$F_2(x) = \frac{1}{0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}} \tag{5}$$

Where a_{1j} and a_{2j} is:

$$a_{1j} = \begin{cases} -32 \rightarrow \mathrm{mod}(j,25) = 1 \\ -16 \rightarrow \mathrm{mod}(j,25) = 2 \\ 0 \rightarrow \mathrm{mod}(j,25) = 3 \\ 16 \rightarrow \mathrm{mod}(j,25) = 4 \\ 32 \rightarrow \mathrm{mod}(j,25) = 5 \end{cases}$$

$$a_{2j} = \begin{cases} -32 \rightarrow j > 0 \wedge j \leq 5 \\ -16 \rightarrow j > 5 \wedge j \leq 10 \\ 0 \rightarrow j > 10 \wedge j \leq 15 \\ 16 \rightarrow j > 15 \wedge j \leq 20 \\ 32 \rightarrow j > 20 \wedge j \leq 25 \end{cases}$$

The third test function is the generalised Griewangk function:

$$F_3(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod \cos(\frac{x_i}{\sqrt{i}}) + 1 \tag{6}$$

The fourth function is the Sphere function:

$$F_4(x) = \sum_{i=1}^{n} x_i^2 \tag{7}$$

The fifth function is the Ackley function:

$$F_5(x) = 20 + e - 20e^{-0.2\left[\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right]} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)} \tag{8}$$

The sixth function is the Step function:

$$F_6(x) = 6n + \sum_{i=1}^{n} \lfloor x_i \rfloor \tag{9}$$

The final function is the Schwefel's Double Sum function:

$$F_7(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2 \tag{10}$$

Table 1 and Figures (1-7) show the results of SUPER-SAPSO in comparison with SAPSO and PSO algorithms respectively.

In all of examinations the numbers of particles are the same and equal to 30. Each examination is repeated 20 times and the average value is reported.

Table 1 Performance results of SUPER-SAPSO, SAPSO and PSO algorithms on benchmark functions

FUNCTION NAMES	ALGORITHM	DIMENSIONS (n)	NUMBER OF ITERATIONS	AVERAGE ERROR
Rastrigin	PSO	20	2989	0.097814
	SAPSO		2168	0.07877
	SUPER-SAPSO		5	0.0
Foxholes	PSO	2	652	0.004846
	SAPSO		536	0.002848
	SUPER-SAPSO		6	0.0
Griewangk	PSO	20	2004	0.082172
	SAPSO		1517	0.075321
	SUPER-SAPSO		3	0.0
Sphere	PSO	20	805	0.094367
	SAPSO		503	0.085026
	SUPER-SAPSO		4	0.0
Ackley	PSO	20	4909	0.099742
	SAPSO		3041	0.099461
	SUPER-SAPSO		5	0.0
Step	PSO	20	10	0.0
	SAPSO		8	0.0
	SUPER-SAPSO		3	0.0
SchwefelDoubleSum	PSO	20	1964	0.086542
	SAPSO		847	0.074965
	SUPER-SAPSO		4	0.0

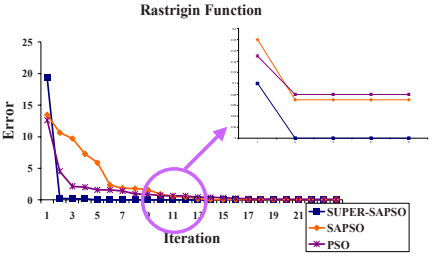


Fig. 1 Convergence comparison of SUPER-SAPSO, SAPSO and PSO algorithms on Rastrigin function

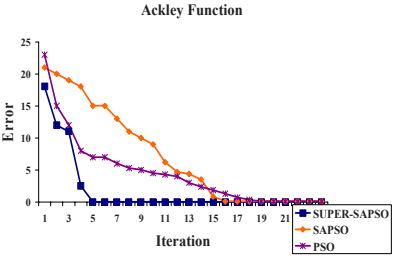


Fig. 2 Convergence comparison of SUPER-SAPSO, SAPSO and PSO algorithm on Ackley function

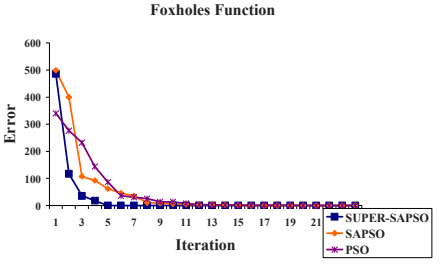


Fig. 3 Convergence comparison of SUPER-SAPSO, SAPSO and PSO algorithms on Foxholes function

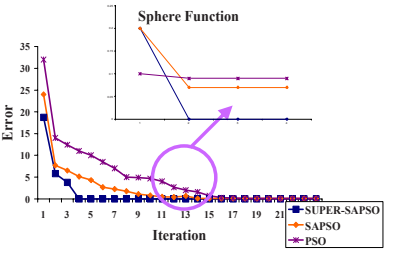


Fig. 4 Convergence comparison of SUPER-SAPSO, SAPSO and PSO algorithms on Sphere function

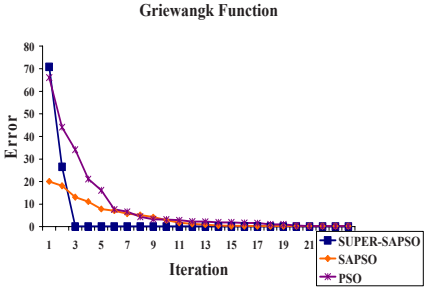


Fig. 5 Convergence comparison of SUPER-SAPSO, SAPSO and PSO algorithms on Griewangk function

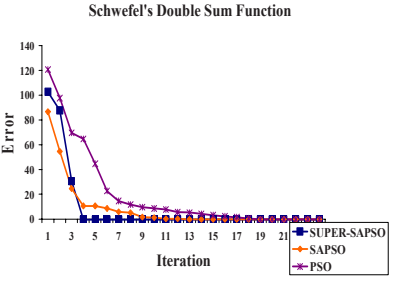


Fig. 6 Convergence comparison of SUPER-SAPSO, SAPSO and PSO algorithms on Schwefel's Double Sum function

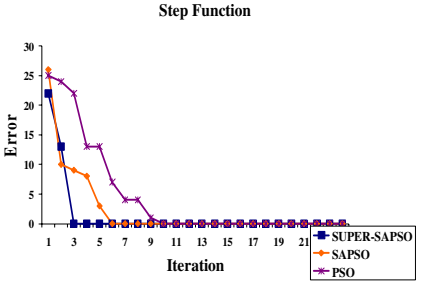


Fig. 7 Convergence comparison of SUPER-SAPSO, SAPSO and PSO algorithms on Step function

6 Conclusions and Future Works

In this paper a new SA-based PSO algorithm, namely SUPER-SAPSO, is proposed. Various tests are carried out on different bench mark functions and

superiority of the proposed model demonstrated. The empirical results show that convergence ratio of SUPER-SAPSO algorithm is almost 282 times faster than SAPSO and 435 times faster than standard PSO algorithm in average. The proposed model not only augments convergence speed but it reduces the optimisation error as well. Therefore the proposed algorithm can move the particles faster towards global optimum with bearing less error.

The results are promising; we are about to develop appropriate multi-objective version of SUPER-SAPSO and solve some applied engineering problems with SUPER-SAPSO and report the results in near future.

References

1. Kennedy, J., Eberhart, R.: Swarm Intelligence. The Morgan Kaufmann Series in Evolutionary Computation (2000)
2. Wang, X.-H., Li, J.-J.: Hybrid Particle Swarm Optimization with Simulated Annealing. In: Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai (2004)
3. Bayraktar, Z., Werner, P.L., Werner, D.H.: Array Optimization via Particle Swarm Intelligence. In: Antennas and Propagation Society International Symposium (2005)
4. Russell, S.J., Norving, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Pearson Education, Inc., London (2003)
5. Eglese, R.W.: Simulated Annealing: A Tool for Operational Research. European Journal of Operational Research 76(3), 271–281 (2000)
6. Ingber, L.: Simulated Annealing: Practice Versus Theory. Mathl. Comput. Modelling (2001)
7. Oliveira, L.S.: Proving Cascading Classifiers with Particle Swarm Optimization. In: Eighth International Conference on Document Analysis and Recognition (ICDAR 2005) (2005)
8. Oliveira, L.S., Britto, A.S., Sabourin Jr., R.: Optimizing Class-related Thresholds with Particle Swarm Optimization. In: 2005 IEEE International Joint Conference on IJCNN 2005 Proceedings (2005)

Testing of Diversity Strategy and Ensemble Strategy in SVM-Based Multiagent Ensemble Learning

Lean Yu, Shouyang Wang, and Kin Keung Lai

Abstract. In this study, a four-stage SVM-based multiagent ensemble learning approach is proposed for group decision making problem. In the first stage, the initial dataset is divided into training subset and testing subset for training and testing purpose. In the second stage, different SVM learning paradigms with much dissimilarity are constructed as diverse agents for group decision making. In the third stage, multiple single SVM agents are trained using training subset and the corresponding decision resulcts are also obtained. In the final stage, all individual results produced by multiple single SVM agents are aggregated into a group decision result. Particularly, the effects of different diversity strategies and different ensemble strategies on multiagent ensemble learning system are tested. For illustration, one credit application approval dataset is used and empirical results demonstrated the impacts of different diversity strategies and ensemble strategies.

1 Introduction

Multiagent ensemble learning has been turned out to be an efficient learning paradigm to achieve high performance, especially in fields where the development of a powerful single learning system requires considerable efforts [12]. Recently, multiagent ensemble learning has been applied for group decision making (GDM) and provided good decision support for multicriteria GDM problems [11]. In the context of multiagent systems, an agent is usually defined as an independent

Lean Yu · Shouyang Wang
Institute of Systems Science, Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing 100190, China
e-mail: {yulean,sywang}@amss.ac.cn

Kin Keung Lai
Department of Management Sciences, City University of Hong Kong, 83 Tat Chee Avenue,
Kowloon, Hong Kong
e-mail: mskklai@cityu.edu.hk

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 431–440.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

decision unit participating as a member of a GDM process. Generally, multiagent ensemble learning can be divided into two categories: competitive ensemble learning, where agents work asynchronously on the same problem and the decision of best agent is the group decision, and cooperative ensemble learning, where the group decision is a fusion or aggregation of the individual decisions of all agents. However, an effective ensemble learning system may not be an individual model but the combination of several of them from a decision support system perspective, according to Olmeda and Fernandez [5]. Usually, ensemble learning model outperforms the individual learning models, whose performance is limited by the imperfection of feature extraction, learning algorithms, and the inadequacy of training data. Another reason supporting this argument is that different individual learning models have their inherent drawbacks and thus aggregating them may lead to a good learning system with high generalization capability. From the above descriptions, we can conclude that there are two essential requirements for ensemble learning. The first is that the ensemble members or agents must be diverse or complementary, i.e., agents must show different properties. Another condition is that an optimal ensemble strategy is also required to fuse diverse agents into an aggregated result.

To achieve high performance, this study attempts to utilize a highly competitive machine learning paradigm – support vector machine (SVM) first proposed by Vapnik [9] – as a generic agent for ensemble learning. The main reasons of selecting SVM as ensemble learning agent reflect the following three folds. First of all, SVM requires less prior assumptions about the input data, such as normal distribution and continuity. Second, SVM can perform a nonlinear mapping from an original input space into a high dimensional feature space, in which it constructs a linear discriminant function to replace the nonlinear function in the original low dimension space. This characteristic help solve the dimension disaster problem because its computational complexity is not dependent on sample dimension. Third, SVM uses structural risk minimization (SRM) to maximize the classification margin, therefore possessing good generalization ability. This pattern can directly help SVM escape local minima, which are often occurred in artificial neural networks (ANN) [9]. These characteristics make SVM popular in many applications.

The main purpose of this paper is to design a SVM-based multiagent ensemble learning system for GDM problem and meantime test the impacts of different diversity strategies and ensemble strategies on SVM-based multiagent ensemble learning system. The rest of the paper is organized as follows. The next section presents a formulation process of a multistage SVM-based multiagent ensemble learning system in detail. For illustration, a credit evaluation example is conducted and the experimental results are reported in Section 3. Section 4 concludes this paper.

2 Methodology Formulation

In this section, a four-step SVM-based multiagent learning system is proposed for GDM problem. First of all, the initial dataset is divided into two independent subsets: training subset (in-sample data) and testing subset (out-of-sample data) for

learning and testing purpose. Particularly, validation subset is also included in the training subset by *k*-fold cross-validation (CV) approach. Then diverse SVM models are constructed as learning agents for group decision making. Subsequently, multiple individual SVM agents are trained using training subset and the corresponding decision results are obtained. Finally, all individual results produced by multiple single SVM agents are aggregated into a group decision result.

Step I: Data Partition. In applications of SVM on group decision analysis, data partition is necessary. Some empirical results (e.g., [2] and [4]) shown that data division can have a significant impact on the results. Generally, data partition is carried out on an arbitrary basis and the statistical properties of the data were seldom considered. In most cases, two subsets, i.e., training subset and testing subset, are used. In this study, the overall data is also divided into two subsets. Particularly, 80% data are used as the training subset and the remainder as the testing subset.

Step II: Diverse SVM Agent Creation. In order to capture the implicit patterns hidden in the data from different perspectives, diverse SVM agents should be used. Generally, an effective multi-agent ensemble learning system consisting of diverse models with much disagreement is more likely to have a good performance in terms of the principle of bias-variance trade-off [13]. Therefore, how to produce diverse models is a crucial factor for building an effective multi-agent ensemble learning system. For SVM models, several diversity strategies have been investigated for the generation of ensemble members. Such diversity strategies basically rely on varying the training samples and parameters related to the SVM design. In particular, some main diversity strategies include the following three aspects:

(1) Data diversity. Because different data often contain different information, these different data can generate diverse models with the capability of capturing different patterns. Usually, in machine learning models, training data is used to construct a concrete model and thus different training data will be an important source of diversity if the training data at hand is functionally or structurally divisible into some distinct training subsets. In order to achieve diverse models, some typical data sampling approaches, such as bootstrap aggregating (bagging) proposed by Breiman [1] and noise injection [7], are used to create data diversity. In this study, the bagging algorithm is adopted as a data sampling tool.

(2) Parameter diversity. In SVM model, there are two typical parameters: regularization parameter C and kernel parameters. By changing the SVM model parameters, different SVM models with much disagreement can be created.

(3) Kernel diversity. In SVM model, kernel function has an important impact on SVM performance. Hence, using different kernel functions in SVM model can also create diverse SVM models. In SVM model, polynomial function, sigmoid function, and RBF function, are typical kernel functions. In this study, RBF kernel is used.

Step III: Single SVM Agent Learning. After creating diverse SVM models, the next step is to train the diverse SVM agents using the training data to create different decision results. In this study, the Vapnik's SVM [9] is used as the single agent.

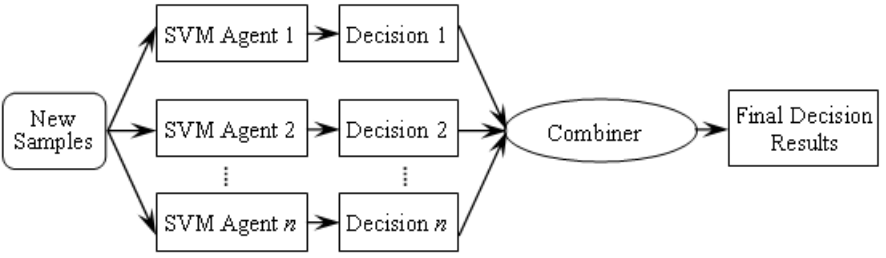


Fig. 1 Multi-agent ensemble learning mechanism for group decision making

Step IV: Multiagent Ensemble Learning. After single SVM agents are generated, each SVM agent can output its own decision results. Although one agent based on a single SVM model may have good performance, it is sensitive to samples and parameter settings, i.e. different SVM agent may have bias. One effective way to reduce the bias is to integrate these SVM agents into an aggregated output for GDM. Fig. 1 shows the multi-agent ensemble learning mechanism for GDM purpose.

From the previous descriptions and Fig. 1, it is easy to find that how to construct a combiner, and how to create the diversity, are two main factors to build an effective ensemble learning system. Because the previous steps have produced some diverse SVM agents in terms of different diversity strategies, how to construct a combiner has become a key factor to construct an effective ensemble learning system.

However, before integrating these SVM agents, strategies of selecting SVM agents must be noted. Generally, these strategies can be divided into two categories: (i) generating an exact number of SVM agents; and (ii) overproducing SVM agents and then selecting a subset of these agents [10].

For the first strategy, several ensemble approaches, e.g., boosting [7], can be employed to generate the exact number of diverse ensemble members for ensemble purpose. Therefore, no selection process will be used and all generated ensemble agents will be combined into an aggregated output. For the second strategy, its main aim is to create a large set of agent candidates and then choose some most diverse agents for ensemble purpose. The selection criterion is some error diversity measures, which is introduced in detail by Partridge and Yates [6]. Because the first strategy is based on the idea of creating diverse SVMs at the early stage of design, it is better than the second strategy, especially for some situations where access to powerful computing resources is restricted. The main reason is that the second strategy cannot avoid occupying much computing time and storage space while creating a large number of agents, some of which are to be later discarded [10].

Generally, there are some ensemble strategies to consider different decision opinions in the existing studies. Typically, majority voting and weighted averaging are two popular ensemble strategies. The main idea of majority voting strategy is to take the vote of the majority of the population of agents. To our knowledge, majority voting is the most widely used ensemble strategy due to its easy implementation. In the majority voting strategy, each agent has the same weight and the voting of the ensemble agents will determine the final decision. Usually, it takes over half the

ensemble to agree a result for it to be accepted as the final output of the ensemble regardless of the diversity and accuracy of each model's generalization. In the mathematic form, the final decision can be represented as

$$F(x) = sign\left(\sum_{i=1}^{n} f_i(x)/n\right), \tag{1}$$

where $f(x)$ is the output of each SVM agent, $F(x)$ is the aggregated output, n is the number of SVM agents. Although this strategy is easy to use, a deadly problem of this strategy is that it ignores the fact some SVM agents that lie in a minority sometimes do produce the more accurate results than others. At the ensemble stage, it ignores the existence of diversity that is the motivation for ensemble learning [10].

Weighted averaging is where the final decision is calculated in terms of individual agents' performance and a weight attached to each agent's output. The gross weight is one and each agent is entitled to a portion of this gross weight based on their performance. A typical binary classification example is to use validation examples to determine the performance weight. Suppose that AB represents the number of observed Class A instances that are mistakenly classified as Class B by an agent, AA denotes the number of correctly classified Class A instances that belong to Class A; BA represents the number of observed Class B instances that are mistakenly classified as Class A, while BB represents the number of correctly classified Class B instance that belong to Class B. Some common measures used to evaluate the performance of the agent are defined below:

$$\text{Type I Accuracy} = \text{Specificity} = \frac{BB}{BB + BA}, \tag{2}$$

$$\text{Type II Accuracy} = \text{Sensitivity} = \frac{AA}{AA + AB}, \tag{3}$$

$$\text{Total Accuracy (TA)} = \frac{AA + BB}{AA + AB + BB + BA}. \tag{4}$$

Each SVM agent is trained by a training subset and is verified by the validation samples. Assume the total accuracy on validation samples of agent i is denoted by TA_i, the weight of agent i denoted by w_i can be calculated as

$$w_i = \frac{TA_i}{\sum_{i=1}^{n} TA_i}. \tag{5}$$

Then the final group decision value of this strategy is shown as follows:

$$F(x) = sign\left(\sum_{i=1}^{n} [f_i(x) \cdot w_i]\right). \tag{6}$$

The above *TA*-based weight averaging ensemble strategy is a class of typical ensemble strategy, the weight determination is heavily depended on the validation samples. When there is no validation subset in data division, this ensemble strategy will be useless. For this reason, this study attempts to propose an adaptive weight determination method based on an adaptive linear neural network (ALNN) model.

The ALNN is a single-layer neural network, where the transfer function is a pure linear function, and the learning rule is Widrow-Hoff (i.e. least mean squared (LMS) rule) [3]. Typically, the mathematical form is expressed by

$$F(x) = \varphi(\sum_{i=1}^{m} w_i f(x_i) + b), \tag{7}$$

where $f(x_i)$ $(i = 1, 2, \ldots, m)$ represents the input variables, $F(x)$ is the output, b is a bias, $w_i (i = 1, 2, \ldots, m)$ is the connection weight, m is the number of input nodes, and $\phi(\cdot)$ is the transfer function, which is determined by the ALNN.

3 Experiment Study

3.1 Data Description and Experiment Design

In this subsection, an England credit application approval dataset is used. It is from a financial service company of England, obtained from accessory CDROM of Thomas et al. [8]. The dataset includes detailed information of 1225 applicants, including 323 observed bad applicants. In the 1225 applicants, the number of good cases (902) is nearly three times that of bad cases (323). To make the numbers of the two classes near equal, we triple the number of bad cases, i.e. we add two copies of each bad case. Thus the total dataset grows to 1871 cases. The purpose of doing this is to avoid having too many good cases or too few bad cases in the training samples. Then we randomly draw 1000 cases comprising 500 good cases and 500 bad cases from the total of 1871 cases as the training samples and treat the rest as testing samples (i.e. 402 good applicants and 469 bad applicants). In these cases, each case is characterized by 14 decision attributes, which are described in [8].

Using this dataset and previous descriptions, we can construct a practical SVM-based multiagent ensemble learning system for GDM. The basic purpose of the multiagent ensemble learning model is to make full use of knowledge and intelligence of each agent in the multiagent system. In the multiagent decision making system, intelligent techniques rather than human experts are used as agents, which is different from traditional multiagent system. In this study we use SVM with much dissimilarity as learning agents for the purpose of GDM.

Subsequently, in order to investigate the impacts of diversity strategy and ensemble strategy on the performance of multiagent ensemble learning system, we perform different experiments with different diversity strategies and ensemble strategies. When testing the effects of diversity strategies on performance of multiagent ensemble learning system, the ensemble strategy is assumed to be fixed. Similarly, the diversity strategy will be fixed if the effects of ensemble strategies on performance of multiagent ensemble learning system are tested.

In the testing of diversity strategies, majority voting based ensemble strategy is adopted. For data diversity, bagging algorithm is utilized. In this study, we use 5 different training scenarios for SVM learning. In this case, the SVM model with RBF kernel is used as the learning agent. The parameters of SVM are determined by grid

search. For parameter diversity, the upper bound parameter C and RBF kernel parameter σ will be varied. In this case, the SVM model with the RBF kernel is used as the learning agent. The training data used is the initial training data produced by data division. For kernel diversity, three typical kernel function, polynomial function, RBF function and sigmoid function are used. In this case, the parameters of SVM are determined by grid search. The training data used is the initial training data produced by data division.

In the testing of ensemble strategies, bagging-based diversity strategy is used. In this case, the SVM model with Gaussian kernel is used as intelligent learning agent. The parameters of SVM are determined by grid search.

To guarantee the robustness of the SVM model, the two-fold cross validation (CV) method is used. In the two-fold CV method, the first step is to divide the training dataset into two non-overlapping subsets. Then we train a SVM using the first subset and validate the trained SVM with the second subset. Subsequently, the second subset is used for training and the first subset is used for validation. Use of the two-fold CV is actually a reasonable compromise, considering the computational complexity of the SVM learning paradigms. Furthermore, an estimate from the two-fold CV is likely to be more reliable than an estimate from a common practice using a single validation set. Besides model robustness, validation sample of this experiment is also used to find weights for different agents for ensemble purpose. Typically, three evaluation criteria, Type I Accuracy, Type II Accuracy and Total Accuracy are used to measure classification results.

3.2 Experiment Results

In this subsection, four different experiments about data diversity, parameter diversity, kernel diversity, and ensemble strategy are conducted. In the data diversity testing, the main goal is to test the effect of the number of training samples on the generalization performance of the multiagent ensemble learning model. For this purpose, we adopt the bagging algorithm to create 500, 1000, 2000 and 5000 different training samples (4 scenarios), as mentioned previous. Using different training samples, different single SVM models with different contents can be produced. With the trained SVM models, some classification results can be achieved. Using the majority voting strategy, the final multiagent group decision results are obtained from multiagent ensemble learning system. Accordingly the final results are shown in Table 1. Note that the values in brackets are standard deviations.

As can be seen from Table 1, two conclusions can be easily found. On the one hand, as the increase of the number of training samples, the performance improvements are generally increased. On the other hand, performance improvement degree from 500 samples to 1000 samples are larger than the performance improvement degree from 1000 samples to 5000 samples, which indicate that the performance improvement degree is not fully dependent of the number of training samples.

In the parameter diversity testing, the SVM model with Gaussian kernel is used as intelligent learning agent, as specified in Section 3.1. Therefore different upper

Table 1 Performance comparisons of SVM ensemble learning with data diversity

Training Samples	Type I (%)	Type II (%)	Total Accuracy (%)
500	69.38 [3.23]	66.36 [3.71]	67.82 [3.52]
1000	73.44 [2.81]	67.93 [3.26]	70.54 [3.03]
2000	73.97 [2.76]	68.14 [3.15]	70.91 [2.87]
5000	73.65 [2.73]	68.72 [3.29]	71.06 [2.85]

bound parameter C and kernel parameter σ are used to create diversity. In particular, the upper bound parameter C varies from 10 to 90 with step size of 20 and the kernel parameter σ changes from 10 to 90 with step size of 20. Some computational results are shown in Table 2.

Table 2 Performance Comparisons of SVM ensemble learning with different parameters

C	σ	Type I (%)	Type II (%)	Total Accuracy (%)
10	10, 30, ..., 90	67.22 [4.31]	63.45 [3.76]	65.11 [3.88]
30	10, 30, ..., 90	68.45 [3.43]	64.89 [3.87]	66.43 [3.63]
50	10, 30, ..., 90	70.08 [3.24]	65.23 [4.54]	67.62 [3.93]
70	10, 30, ..., 90	71.54 [3.98]	67.46 [4.22]	69.54 [4.07]
90	10, 30, ..., 90	70.12 [4.02]	68.82 [4.65]	69.21 [4.36]
10, 30, ..., 90	10	69.43 [4.13]	67.56 [3.78]	68.74 [4.04]
10, 30, ..., 90	30	68.45 [4.34]	66.19 [3.98]	67.18 [4.17]
10, 30, ..., 90	50	68.34 [4.28]	64.45 [4.34]	66.78 [4.32]
10, 30, ..., 90	70	67.23 [4.54]	66.68 [3.56]	66.87 [4.03]
10, 30, ..., 90	90	66.43 [4.38]	65.67 [4.78]	65.94 [4.56]

From Table 2, we can find that different parameter settings can lead to different generalization performance, but the differences are insignificant. Concretely speaking, the classification performance on the testing data increases when C increases from 10 to 70, but classification performance decreases when C increases from 70 to 90. Thus a suitable upper bound parameter is of utmost importance to SVM learning. The main reason is that the regularized parameter C is an important trade-off parameter between margin maximization and tolerable classification errors. If the selection of regularized parameter C is inappropriate, it might lead to an unexpected generalization results. However, for kernel parameter σ^2, it is difficult to find similar results and the possible reason is worth further exploring in the future research.

Similarly, we can also use different kernel functions to create multiagent SVM ensemble learning system for group decision making. For this purpose, three different kernel functions: polynomial function, sigmoid function and RBF function are used for testing. In order to construct a diverse ensemble learning system, 1000 copies of the same training dataset are replicated. In the 1000 training samples, different kernel functions are hybridized to create the kernel diversity. Note that in every experiment only one kernel function dominates the multiagent ensemble learning system. Detailed configuration of the kernel functions and corresponding computational results are shown in Table 3.

Table 3 Performance comparisons of SVM ensemble learning with kernel diversity

Kernel Functions	Type I (%)	Type II (%)	Total Accuracy (%)
600Poly+200Sig+200RBF	64.76 [4.76]	61.89 [5.05]	63.25 [4.94]
200Poly+600Sig+200RBF	68.14 [4.28]	65.51 [4.69]	66.68 [4.55]
200Poly+200Sig+600RBF	71.32 [4.93]	68.13 [4.71]	68.59 [4.87]

From Table 3, several interesting results are found. First of all, the multiagent ensemble learning system dominated by Gaussian-type kernel function can produce the best classification results, followed by sigmoid kernel and polynomial kernel. Second, relative to the polynomial kernel function, there is a significant difference at 10% level by two-tail t-test. Third, although the ensemble learning system dominated by RBF kernel has an advantage over the ensemble learning system dominated by sigmoid kernel, but the robustness of the former is slightly worse than that of the latter. The possible reason is unknown, which is worth further exploring.

Subsequently, we mainly test the effects of ensemble strategy on overall performance in multiagent GDM system. As described above, bagging-based diversity strategy is used. Particularly, 1000 training samples are replicated by bagging algorithm in this section. In addition, the SVM models with RBF kernel function are used as the learning agent. The parameters of SVM are determined by grid search. Particularly, three main ensemble strategies: majority voting, TA-based weighted averaging and ALNN-based adaptive weighting, are used for comparison. Note that the same SVM-based classifiers with different training samples are adopted. Accordingly, the computational results are shown in Table 4.

Table 4 Performance comparisons of SVM ensemble model with different ensemble strategies

Ensemble Strategies	Type I (%)	Type II (%)	Total Accuracy (%)
Majority voting	68.63 [5.43]	66.24 [5.72]	67.31 [5.55]
TA-based weighted averaging	69.96 [4.65]	67.45 [4.21]	68.62 [4.37]
ALNN-based adaptive weighting	72.52 [4.63]	69.98 [4.44]	71.19 [4.53]

As can be seen from Table 4, we can find that (1) the ALNN-based adaptive weighting approach to ensemble learning can produce the best result among the three ensemble strategies listed in this study. (2) Of the three ensemble strategies, the majority voting strategy is the worst. The main reason is that that it ignores the existence of diversity that is the motivation for multi-agent ensemble learning system [9]. (3) Performance improvement (71.19-68.62) of the ALNN-based adaptive weighting ensemble strategy relative to TA-based weighted averaging ensemble strategy is much better than that (68.62-67.31) of the TA-based weighted averaging ensemble strategy relative to majority voting ensemble strategy. (4) Relative to the majority voting ensemble strategy, there is a significant difference by two-tail t-test,

which indicates that the ALNN-based adaptive weighting ensemble learning system is a promising ensemble strategy for multiagent ensemble learning system.

4 Conclusions

In this study, a multi-stage SVM-based multiagent ensemble learning paradigm is proposed for group decision making problem. Empirical results show that different diversity strategies and different ensemble strategies can produce different impact on performance of group decision making problem. Particularly, data diversity is better than other two diversity strategies. Meantime, ALNN-based ensemble strategy is better than other weighting strategies. These interesting findings provide a useful guidance to construct an efficient SVM-based multiagent ensemble learning system.

Acknowledgements

This work is partially supported by grants from the National Natural Science Foundation of China (No. 70221001), the Knowledge Innovation Program of the Chinese Academy of Sciences, and the NSFC/RGC Joint Research Scheme (No. N_CityU110/07).

References

1. Breiman, L.: Bagging predictors. Machine Learning 26, 123–140 (1996)
2. Dawson, C.W., Wilby, R.: An artificial neural network approach to rain-fall-runoff modeling. Hydrological Sciences Journal 43, 47–66 (1998)
3. Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural Network Design. PWS Publishing Company, Boston (1996)
4. Maier, H.R., Dandy, G.C.: Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications. Environmental Modelling & Software 15, 101–124 (2000)
5. Olmeda, I., Fernandez, E.: Hybrid Classifiers for Financial Multicriteria Decision Making: The Case of Bankruptcy Prediction. Computational Economics 10, 317–335 (1997)
6. Partridge, D., Yates, W.B.: Engineering multiversion neural-net systems. Neural Computation 8, 869–893 (1996)
7. Schapire, R.E.: The strength of weak learnability. Machine Learning 5, 197–227 (1990)
8. Thomas, L.C., Edelman, D.B., Crook, J.N.: Credit Scoring and Its Applications. Society of Industrial and Applied Mathematics, Philadelphia (2002)
9. Vapnik, V.: The nature of statistical learning theory. Springer, New York (1995)
10. Yang, S., Browne, A.: Neural network ensembles: combining multiple models for enhanced performance using a multistage approach. Expert Systems 21, 279–288 (2004)
11. Yu, L., Wang, S.Y., Lai, K.K.: An Intelligent-Agent-Based Fuzzy Group Decision Making Model for Financial Multicriteria Decision Support: The Case of Credit Scoring. European Journal of Operational Research (2007), doi:10.1016/j.ejor.2007.11.025
12. Yu, L., Wang, S.Y., Lai, K.K.: Credit risk assessment with a multistage neural network ensemble learning approach. Expert Systems with Applications 34(2), 1434–1444 (2008)
13. Yu, L., Wang, S.Y., Lai, K.K., Huang, W.: A bias-variance-complexity trade-off framework for complex system modeling. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3980, pp. 518–527. Springer, Heidelberg (2006)

Probability Collectives: A Decentralized, Distributed Optimization for Multi-Agent Systems

Anand J. Kulkarni and K. Tai

Abstract. Complex systems may have many components that not only interact but also compete with one another to deliver the best they can to reach the desired system objective. As the number of components grows, complexity and communication also grow, making them computationally cumbersome to be treated in a centralized way. It may be better to handle them in a distributed way and decomposed into components/variables that can be seen as a collective of agents or a Multi-Agent System (MAS). The major challenge is to make these agents work in a coordinated way, optimizing their local utilities and contributing towards optimization of the global objective. This paper implements the theory of Collective Intelligence (COIN) using Probability Collectives (PC) in a slightly different way from the original PC approach to achieve the global goal. The approach is demonstrated successfully using Rosenbrock Function in which the variables are seen as agents working independently but collectively towards a global objective.

1 Introduction

The desired system objective of a complex system is achieved by its various components delivering their best by interacting and competing with one another. Traditionally, the complex systems were seen as centralized systems, but as the complexity grew, it became necessary to handle the systems using distributed and decentralized optimization. In a decentralized approach, the system can be divided into smaller subsystems and optimized individually to get the system level optimum.

In the approach presented in this paper, the system was divided into subsystems and each subsystem was considered as an agent. The framework of Collective Intelligence (COIN) is an approach to design a collective, which includes the computational agents working to enhance the system performance. Essentially, in COIN approach, the agents select actions over a particular range and receive some rewards on the basis of the system objective achieved because of those actions. The process iterates and reaches equilibrium when no further increase in reward is possible for the individual agent by changing the actions further. This equilibrium concept is known as Nash Equilibrium [1]. This concept can be successfully formalized and implemented through the concept of Probability Collectives (PC). This

Anand J. Kulkarni · K. Tai
School of Mechanical and Aerospace Engineering, Nanyang Technological University,
50 Nanyang Avenue, Singapore 639798, Singapore
e-mail: {kulk0003, mktai}@ntu.edu.sg

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 441–450.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

approach works on probability distribution, directly incorporating uncertainty. The PC approach was implemented by the authors of this paper in their earlier work [2]. The modified PC approach is presented here.

The paper is organized as follows. Section 2 presents an overview of some application areas of distributed Multi-Agent System (MAS). A review of COIN using PC theory and the relevant concepts are given in Section 3. It is followed by the formulation of PC theory in Section 4. Section 5 illustrates the theory using the test problem, followed by results and discussion. Some concluding remarks and future work are given in Section 6.

2 Distributed, Decentralized, Cooperative MAS

Complex adaptive systems characterized by the interaction of agents benefit from highly distributed and parallel processes that lead to enhanced robustness and computational speed [3]. This is the major reason why MASs have been successfully applied to various engineering applications such as engineering design, manufacturing, scheduling, logistics, sensor networks, vehicle routing, UAV path planning and many more are currently being researched.

In the approach presented by [3], the agents in MASs handle the pre- and post-processing of various computational analysis tools such as spreadsheets or CAD systems in order to have a common communication between them. These agents communicate through a common framework where they act as experts and communicate their results to the centralized design process. As the number of design components increase, the number of agents and the complexity also increases. This results in the growing need for communication, making it computationally cumbersome for a centralized system. This is one of the reasons that centralized approach is becoming insignificant in the concurrent design.

Designing a MAS to perform well on a collective task is non-trivial. A case study on token retrieval problem is demonstrated by [4] and showed that the straightforward agent learning in MAS cannot be implemented as it can lead to suboptimal solution and also may interfere with individual interests. This is referred to as 'Tragedy of Commons'. It also highlights the inadequacy of classic Reinforcement Learning (RL) approach. The attractive option is to devise a distributed system in which different parts are referred to as agents which contribute towards a common aim.

A very different distributed MAS approach is illustrated in [5] to provide a monetary conflict-resolution mechanism to accommodate the project schedule changes referred to as Distributed Coordination Framework for Project Schedule Changes (DCPSC). The objective is the minimization of the total extra cost each sub-contactor has to incur. The authors refer every sub-contractor as a software agent to enhance communication using internet. The agents in this distributed system compete and take socially rational decision to maintain logical sequence of the network.

In [6], supply chain management system is demonstrated using COIN approach. Various components in the supply chain act as agents. Some of them are Ware-house Agents, Manufacturer agents, Retailer Agents, etc. These agents are

assigned action sets. Depending upon the relationship between the agents, environmental states and other agents' actions, each individual agent selects appropriate action to fulfill individual goal towards the optimum. Supply chain system is also addressed using a distributed MAS approach of Ant Colony Optimization [7]. The results clearly show that distributed MAS outperforms centralized approach.

Area of Unmanned Vehicles (UVs) or Unmanned Aerial Vehicles (UAVs) is a potential application which can be implemented in decentralized and distributed way. Mainly, this research is for conflict resolution, collision avoidance, better utilization of air-space and mapping efficient trajectory [8], [9].

3 Collective Intelligence (COIN) Using Probability Collectives (PC) Theory

3.1 Probability Collectives (PC)

A collective is a group of learning agents, which are self interested and work in some definite direction to optimize the local rewards or payoffs, which also optimizes the global or system objective. The system objective is also referred to as world utility and is a measure of performance of the whole system. Probability Collectives (PC) theory is a broad framework for modeling and controlling distributed systems, and it has deep connections to Game Theory, Statistical Physics, and Optimization [10].

The method of PC theory is an efficient way of sampling the joint probability space, converting the problem into the convex space of probability distribution. PC considers the variables in the systems as individual agents/players of a game being played iteratively [11]. Unlike stochastic approaches such as Genetic Algorithms (GA), Swarm Optimization and Simulated Annealing (SA), rather than deciding over the agent's moves/set of actions, PC allocates probability values to each agent's moves. In each iteration, every agent independently updates its own probability distribution to select a particular action out of its strategy set having the highest probability of optimizing its own utility (private utility) which also results in optimizing the world utility or system objective [10]. This is based on the prior knowledge of the actions/strategies of all other agents.

3.2 Advantages of PC

PC approach has the following advantages over the other tools that can be used in optimizing collectives.

1. PC is a distributed solution approach in which each agent independently updates its probability distribution at any time instance and can be applied to continuous, discrete or mixed variables, etc. [10], [11], [12]. The probability of the strategy set is always a vector of real numbers, regardless of the type of data under consideration. This exploits the techniques of the optimization for Euclidean vectors, such as gradient descent.

2. It is robust in the sense that the cost function (Global Utility function) can be irregular or noisy i.e. can accommodate noisy and poorly modeled problems [10], [13].
3. It provides sensitivity information about the problem in the sense that a variable with a peaky distribution (having highest probability value) is more important in the solution than a variable with a broad distribution i.e. peaky distribution provides the best choice of action that can optimize the global utility [10], [13].
4. The minimum value of the global cost function can be found by considering the maxent Lagrangian equation for each agent (variable) [11].
5. It can include Bayesian prior knowledge [12].
6. It can efficiently handle the problems having large number of variables [13].

These advantages of PC make it a competitive choice over other algorithms. The key concept of PC theory is that the minimum value of the world utility can be found by considering the maximum entropy (maxent) of agent/variable.

4 COIN Formulation

The detailed implementation of COIN using PC theory is as follows, with the algorithm flowchart represented in Figure 1.

Consider there are N agents. Each agent i is given a strategy set $\mathbf{X}_i$ represented as

$$\mathbf{X}_i = \{X_i^{[1]}, X_i^{[2]}, X_i^{[3]}, ..., X_i^{[m_i]}\} \quad , \qquad i \in \{1, 2, ..., N\} \tag{1}$$

where m_i is the number of strategies for agent i.

1. Assign uniform probabilities to the strategies of Agent i. This is because, at the beginning, the least information is available (largest uncertainty) about which strategy is favorable and results in minimization of the objective function at the particular iteration under consideration of the particular values selected by agents other than i (i.e. guessed by agent i). Therefore, '*at the beginning of the game*', each agent's every strategy has probability $1/m_i$ of being most favorable. Therefore, probability of strategy r of Agent i is

$$q(X_i^{[r]}) = 1/m_i \quad , \qquad r = 1, 2, ..., m_i \tag{2}$$

Agent i selects its first strategy and samples randomly from other agents' strategies as well. This is a random guess by agent i. This forms a '*combined strategy set*' represented as

$$\mathbf{Y}_i^{[r]} = \left\{ X_1^{[r]}, X_2^{[?]}, X_3^{[?]}, ..., X_N^{[?]} \right\} \tag{3}$$

Superscript [?] indicates that it is a '*random guess*' and not known in advance. It is important to note that all the agents make their own '*combined strategy sets*' $\mathbf{Y}_i^{[r]}$.

From each of the '*combined strategy sets*' $\mathbf{Y}_i^{[r]}$ for agent i, compute the '*expected local utility*' for each agent i with strategy i as follows:

$$Exp\ \ Utility\ of\ \ Agent\ \ i^r = q_i^r \prod_{(i)} q(X_{(i)}^{[?]}) \bullet G(\mathbf{Y}_i^{[r]}) \tag{4}$$

where (i) represents every agent other than i and G is World Utility/System Objective.

Compute the Expected Global Utility on the basis of these '*combined strategy sets*' for every agent i. This is problem dependent. Update the probabilities of all the strategies of each agent i using Boltzmann's temperature T as follows,

$$q(X_i^r) \leftarrow q(X_i^{[r]}) - \alpha_{step} \bullet q(X_i^{[r]}) k_{r.update} \tag{5}$$

$$\text{where } k_{r.update} = \frac{(Contribution\ of\ \ Agent\ \ i)}{T} + S_i(q) + \ln(q(X_i^{[r]})) \text{ and} \tag{6}$$

$$Contribution\ \ of\ Agent\ i = Exp\ \ Utility\ \ of\ \ Agent\ i^r\ -\ \ Exp\ Global\ \ Utility \tag{7}$$

2. Using the above *second order update rule*, the strategy having maximum contribution towards the minimization of the objective is separated from the other strategies, i.e. its probability is increased iteratively.

   For any agent i, if strategy r contributes more towards minimization of the objective than other strategies, its probability increases with certain amount greater than the other strategies. This updating continues for predefined number of iterations to get clear probability distribution showing the highest probability for certain strategy value.

   Compute the '*entropy*' of every agent's probability distribution as

$$S_i(q) = -\sum_{r=1}^{m_i} q(X_i^{[r]}).\ln\ q(X_i^{[r]}) \tag{8}$$

   Entropy is the indication of the information availability. It increases when the information becomes clearer and reaches to maximum when information available also reaches to the maximum. When the entropy reaches the maximum, the probability distribution available clearly distinguishes every strategy's contribution towards the minimization of the Expected Global Utility.
   For detailed discussion on entropy refer to [10].

   Repeat the procedure from Step 2 through Step 3 for predefined iterations k.

3. For each agent, identify the strategy, which contributes the maximum in minimizing the Global Utility and refer to it as the '*favorable strategy*'.

   Store the following results:

   1. The '*favorable strategy value*'
   2. The '*sampling matrix*' from which the '*favorable strategy*' achieved

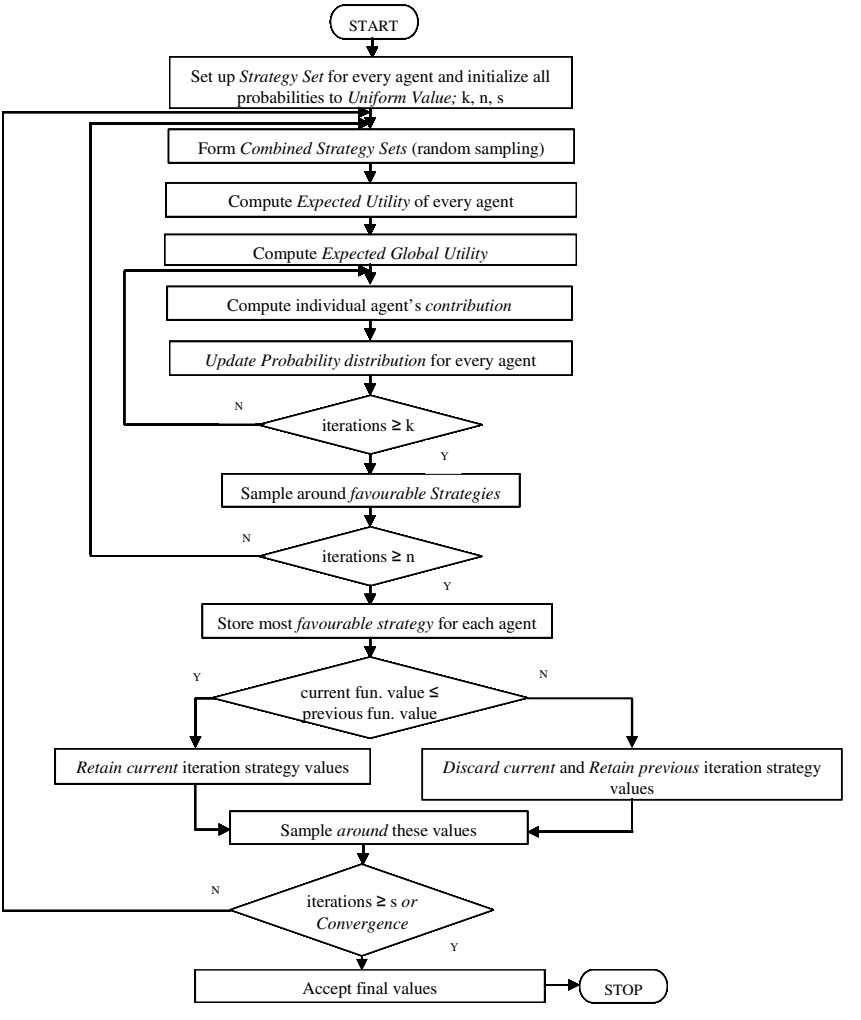


Fig. 1 Algorithm of PC Approach Implemented

Sample around '*favorable strategy*' to select neighboring values within the range,

$$New\ Strategy\ Limits = Favorable\ Strategy \pm (0.5) \times Favorable\ Strategy \quad (9)$$

4. Call the '*most favorable strategy*' out of all stored, along with its corresponding '*sampling matrix*'.

   Repeat the procedure from Step 1 through Step 5 for predefined iterations n.
5. Sample around the '*most favorable result*' to select neighboring values in the reduced range

$$New\ Strategy\ Limits = Favorable\ Strategy \pm (0.1) \times Favorable\ Strategy \quad (10)$$

6. Output the '*recent strategy*' values and the corresponding objective function. Repeat the procedure from Step 1 through Step 7 until the convergence criterion is satisfied or for predefined number of iterations s.

It is worth to mention that the convergence criterion in the present PC approach was the number of predefined iterations and/or there is no change in final goal value for a considerable number of iterations. On the other hand, the convergence criterion in the PC approach originally presented by [12], [13] and also by the authors of this paper in their previous work presented in [2] was number of iterations for which there is no change in the highest probability value of the most favorable strategy. Also, sampling in further stages of the PC algorithm presented here is done using the range selection in the neighborhood of the most favorable value in the particular iteration while the original PC approach uses regression and data aging approach.

5 Results for Rosenbrock Function Using COIN

There are a number of benchmark test functions for contemporary optimization algorithms like GAs and evolutionary computation. The Rosenbrock function is an example of a non-linear function having strongly coupled variables and is a real challenge for any optimization algorithm because of its slow convergence for most of the optimization methods [14], [15]. The Rosenbrock Function with N number of variables is given by

$$f(\mathbf{X}) = \sum_{i=1}^{N-1} \left[100 \left(x_i^2 - x_{i+1} \right)^2 + \left(1 - x_i \right)^2 \right] \quad (11)$$

where $\mathbf{X} = [x_1\ x_2\ x_3 \ldots\ldots x_N]$, lower limit $\le x_i \le$ upper limit and the global minimum is at $f(\mathbf{X}^*) = 0$, $\mathbf{X}^* = [1\ 1\ 1 \ldots\ldots 1]$. A difficulty arises from the fact that the optimum is located in a deep and narrow parabolic valley with a flat bottom. Also, gradient based methods may have to spend a large number of iterations to reach the global minimum.

In the context of COIN, the function variables were seen as a collection of agents with each variable represented as an autonomous agent. These agents competed with one another to optimize their individual values and ultimately the entire function value (i.e. global utility). Each agent randomly selected the strategy within the range of values specified to each agent. Although the optimal value of each and every variable x_i is 1, the allowable range for each variable (agent) was intentionally assigned to be different (as shown in Table 1). To get the refined results and reach the convergence faster, 42 strategies were considered for every agent from within the pre-specified range. The starting probabilities assigned for each strategy selected randomly was uniform i.e. 1/42. The procedure explained in Section 4 was followed to reach the convergence.

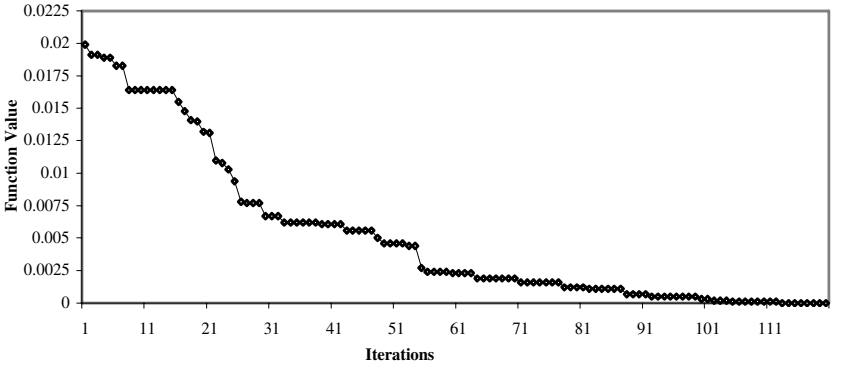


Fig. 2 Results for Trial 5

Table 1 Performance using PC Approach

Agents/ (Variables)	Strategy Values Selected with Maximum Probability					
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Range of Values
Agent 1	1	0.9999	1.0002	1.0001	0.9997	-1.0 to 1.0
Agent 2	1	0.9998	1.0001	1.0001	0.9994	-5.0 to 5.0
Agent 3	1.0001	0.9998	1	0.9999	0.9986	-3.0 to 3.0
Agent 4	0.9998	0.9998	0.9998	0.9995	0.9967	-3.0 to 8.0
Agent 5	0.9998	0.9999	0.9998	0.9992	0.9937	1.0 to 10.0
Fun. Value	2×10^{-5}	1×10^{-5}	2×10^{-5}	2×10^{-5}	5×10^{-5}	
Fun. Eval.	288100	223600	359050	204750	249500	

The problem was coded in MATLAB 7.4.0 (R2007A). The results of 5 trials are shown in Table 1. The results for Trial 5 are plotted in Figure 2. For Trial 5, value of the function at iteration 113 was accepted as the final one as there was no change in the function for considerable number of iterations. The little inconsistency in the function value and the number of function evaluations was due to the randomness in selection of the strategies. A number of researchers have solved the Rosenbrock Function using various algorithms. Table 2 summarizes the results obtained by these various algorithms: Chaos Genetic Algorithm (CGA) [14], Punctuated Anytime Learning (PAL) system [15], Modified Differential Evolution (Modified DE) proposed by [16] and Loosely Coupled GA (LCGA) is implemented by [17] are some of the algorithms demonstrated on the Rosenbrock function.

Referring to Table 2, every variable was assigned the identical range of allowable values, contrary to Table 1 with each variable having different allowable ranges. Further, even with more number of variables, function values are much better in Table 1 compared to Table 2. This makes it clear that the approach

presented here produced fairly good comparable results to those produced by previous researchers.

Furthermore, Monte Carlo sampling used in the original PC approach [11], was computationally expensive and slow as the number of samples may be in thousands or even millions. On contrary, in the modified PC approach presented here, pseudorandom scalar values drawn from uniform distribution were fewer in number. Also, the regression (necessary in the original approach to fit the individual utility inside the individual range domain) was completely avoided here, making the modified approach computationally less expensive. Most significantly, with every iteration the sampling range of individual variables (agents) was narrowed down, ensuring faster convergence and an improvement in efficiency over the original PC approach.

Table 2 Performance Comparison of Various Algorithms solving Rosenbrock Function

Method	No. of Var./Agents	Function Value	Function Evaluations	Var. Range(s)/ Strategy Sets
CGA [14]	2	0.000145	250	-2.048 to 2.048
PAL [15]	2	≈ 0.01	5250	-2.048 to 2.048
	5	≈ 2.5	100000	-2.048 to 2.048
Modified DE [16]	2	1×10^{-6}	1089	-5 to 10
	5	1×10^{-6}	11413	-5 to 10
LCGA [17]	2	≈ 0.00003	--	-2.12 to 2.12

6 Conclusions and Future Work

The methodology successfully demonstrated the application of COIN using PC theory. Authors believe that the algorithm is made simpler by changing the sampling method and convergence criterion. It was also evident that the growing complexity can be handled by decomposing the system into smaller subsystems or agents. The implementation of the Rosenbrock function highlighted that the approach yielded optimum solution for coupled variables. This indicated the further investigation in the fields related to the problem of Traveling Salesman and path planning of the Multiple Unmanned Vehicles (MUVs). Authors also see the potential application in the dynamic fields such as machine shop scheduling, distributed sensor networks and vehicle routing in which every agent has different local utility functions.

Acknowledgments. The authors would like to acknowledge Dr. Rodney Teo and Ye Chuan Yeo (DSO National Laboratories, Singapore) for their useful discussions on this topic, as well as Dr. David H. Wolpert (NASA Ames Research Center, Moffett Field, CA, USA) and Dr. Stefan R. Bieniawski (Boeing Phantom Works, Seal Beach, CA, USA) for their help with the study of some of the Probability Collectives concepts.

References

1. Basar, T., Olsder, G.J.: Dynamic Non-Cooperative Game Theory. Academic Press, London (1995)
2. Kulkarni, A.J., Tai, K.: Probability Collectives for Decentralized, Distributed Optimization: A Collective Intelligence Approach. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 1271–1275 (2008)
3. Liu, H., Abraham, A., Clerc, M.: Chaotic Dynamic Characteristics in Swarm Intelligence. Applied Soft Computing 7, 1019–1026 (2007)
4. Hoen, P.J., Bohte, S.M.: COllective INtelligence with sequences of actions: Coordinating actions in multi-agent systems. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) ECML 2003. LNCS, vol. 2837, pp. 181–192. Springer, Heidelberg (2003)
5. Kim, K., Paulson, B.C.: Multi-Agent Distributed Coordination of Project Schedule Changes. Computer-Aided Civil and Infrastructure Engineering 18, 412–425 (2003)
6. Sheremetov, L., Rocha-Mier, L.: Collective Intelligence as a Framework for Supply Chain Management. In: Second IEEE International Conf. on Intelligent Systems, pp. 417–422 (2004)
7. Silva, C.A., Sousa, J.M., Runkler, T.A., S´a da Costa, J.: Distributed optimization in supply–chain management using ant colony optimization. International Journal of Systems Science 37(8), 503–512 (2006)
8. Chandler, P., Rasumussen, S., Pachter, M.: UAV cooperative path-planning. In: AIAA Guidance, Navigation, and Control Conf. Art. No. 4370 (2000)
9. Sislak, D., Volf, P., Komenda, A., Samek, J., Pechoucek, M.: Agent-based multi-layer collision avoidance to unmanned aerial vehicles. In: International Conf. on Integration of Knowledge Intensive Multi-Agent Systems, Art. No. 4227576, pp. 365–370 (2007)
10. Wolpert, D.H.: Information Theory —– The Bridge Connecting Bounded Rational Game Theory and Statistical Physics. Complex Engineering Systems, 262–290 (2006)
11. Wolpert, D.H., Antoine, N.E., Bieniawski, S.R., Kroo, I.R.: Fleet Assignment Using Collective Intelligence. In: 42nd AIAA Aerospace Science Meeting Exhibit, Reno, Nevada, USA (2004)
12. Wolpert, D.H., Tumer, K.: An introduction to COllective INtelligence. Technical Report, NASA ARC-IC-99-63, NASA Ames Research Center (1999)
13. Bieniawski, S.R.: Distributed Optimization and Flight Control Using Collectives. PhD Dissertation, Stanford University (October 2005)
14. Cheng, C.T., Wang, W.C., Xu, D.M., Chau, K.W.: Optimizing Hydropower Reservoir Operation Using Hybrid Genetic Algorithm and Chaos. Water Resource Management 22, 895–909 (2008)
15. Blumenthal, H. J., Parker, G.B.: Benchmarking Punctuated Anytime Learning for Evolving a Multi-Agent Team's Binary Controllers. In: World Automation Congress (WAC) (2006)
16. `http://www.ies.org.sg/journal/current/v46/v462_3.pdf`
17. Bouvry, P., Arbab, F., Seredynski, F.: Distributed evolutionary optimization, in Manifold: Rosenbrock's function case study. Information Sciences 122, 141–159 (2000)

Shape from Focus Based on Bilateral Filtering and Principal Component Analysis

Muhammad Tariq Mahmood, Asifullah Khan, and Tae-Sun Choi

Abstract. Three dimensional (3D) shape reconstruction of an object, from its two 2D images, is an important aspect in machine vision applications. Shape from focus (SFF) is a passive optical method for 3D shape recovery in which best-focused points are located among the image sequence. However, existing approaches largely rely on gradient-based sharpness measures and thus are noise sensitive. Moreover, these approaches locally compute focus quality by summing focus values within a window and consequently produce coarse surface. This paper introduces a new SFF method based on bilateral filtering (BF) and principal component analysis (PCA). In the first step, a sequence of neighborhood vectors is convolved with BF and then in the second step, PCA is applied on the resultant matrix to transform data into eigenspace. The score for the first component is employed to compute the depth value. The comparative analysis demonstrates the effectiveness of the proposed method.

1 Introduction

Shape from focus (SFF) is a passive optical method for 3D shape reconstruction of an object from its 2D images. SFF has numerous applications in computer vision, range segmentation, consumer video cameras, and video microscopy [1]-[2]. In SFF methods, a sequence of images is acquired either by relocating the object along the optical axis in small steps or by changing the focal length of the lens in the camera. A sharpness criterion or a focus measure is applied on each pixel in every frame for measuring the focus quality. The sharpest pixel among the sequence provides depth information. Further, an approximation method is applied to refine the initial results.

Usually in conventional SFF methods, image volume is convolved with Laplacian or Sobel operators and the focus quality for each pixel is estimated by aggregating resultant values within a small 2D window. The depth is then computed by maximizing the focus quality along the optical axis. The size of window affects the depth map accuracy as well as the computational complexity [3]-[4]. Bilal and Choi [5] suggested 3D window for computing accurate depth

Muhammad Tariq Mahmood · Asifullah Khan · Tae-Sun Choi
School of Information and Mechatronics, Gwangju Institute of Science and Technology,
261 Cheomdan-gwagiro, Buk-gu, 500-712, Gwangju, Republic of Korea
e-mail: {tariq, asifullah,tschoi}@gist.ac.kr

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 453 – 462.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

maps. The performance of these SFF techniques largely depends upon detection of sharp variation in intensities such as the edges using high pass filters. However, a shortcoming of these operators is that the noise may also be enhanced. Therefore, the performance of these methods deteriorates, especially, in noisy environment. Moreover, the summation of convolved values within the window produces a single scalar value for each pixel of the current frame. However, it increases the difference between two adjacent frames and therefore, may produce coarse surface as well as an inaccurate depth map of the object.

In this paper, we introduce a novel method for 3D shape recovery based on bilateral filtering (BF) and principal component analysis (PCA). In the first step, the sequence of neighborhood vectors is convolved with bilateral filter, which enables us to suppress noise and to preserve edges simultaneously. In the second step, instead of local summation, PCA is applied on the resultant matrix to transform the convolved data into eigenspace. The score for the first component is employed to calculate the depth value. The experimental results show that the new method is more accurate and robust when compared with previous methods.

2 Related Work

The first step in SFF algorithms is to apply a focus measure. In literature, many focus measures have been reported in the spatial as well as in the frequency domain [6]. Modified Laplacian (ML), sum modified Laplacian (SML), Tenenbaum focus measure (TEN), and gray level variance (GLV) are well known among them. The performance of ML is deteriorated due to the cancellation of derivatives in the opposite direction. Nayar and Nakagawa [7] proposed SML to overcome this limitation. In SML, an image is convolved with Laplacian operator and the focus value is calculated by summing the resultant absolute values over a small window. Tenenbaum [8] proposed a focus measure by convolving an image with horizontal and vertical Sobel masks and then accumulating the energy of resultant gradient vector components. The GLV focus measure is based on the idea that the sharp pixels occupy high variations in gray level values while low variance belongs to blurred image structure. The focus value is calculated for the central pixel of the window by computing the variance of the intensity values. Few focus measures have also been proposed in discrete cosine transform (DCT) domain. Baina et al. [9] proposed the energy of the AC part as a focus measure. Shen and Chen [10] suggested that the ratio between energies of AC and DC parts is a better choice for measuring the focus quality. Kristan et al. [11] proposed the entropy of the normalized DCT coefficients as a focus measure.

In order to get a refined depth map, an approximation method follows the focus measure. Nayar and Nakagawa [7] proposed SFF method based on Gaussian interpolation. Three focus values near the peak, computed by SML, are fitted to the Gaussian model and the position of the mean value is taken as optimal depth

value. Subbarao and Choi[12] introduced the concept of focused image surface (FIS) and proposed a method to recover an accurate 3D shape by using initial estimate taken through SML. Yun and Choi[13] suggested the estimation of FIS through piecewise curved surface approximation. Bilal and Choi[1] proposed a SFF method using dynamic programming (DP) to obtain an optimal FIS. Malik and Choi[2] applied the fuzzy inference system for computing the depth map. Recently, we applied PCA on the sequence of neighborhood vectors to estimate accurate 3D shape [14]. These methods provide better results when compared with the traditional methods, but are computationally expensive.

3 Proposed Algorithm

3.1 Motivation

The main idea behind our approach is to apply PCA on the filtered data instead of original intensities. PCA transforms the data into eigenspace such that the principal axis is in the direction of maximum variation. Thus, first few components contain more variance whereas noisy information is generally collected in lower order components. However, image pixels acquired from a CCD camera are the convolution of actual pixel values and point spread function (PSF). Therefore, many conventional SFF approaches use high pass filters such as Lalacian and Sobel operators to measure the sharpness and following it with local aggregation of the resultant values. The performance of these operators deteriorates significantly, as they may amplify noise. Therefore, to tackle these shortcomings, we propose a new method based on BF and PCA. BF is a non-linear filter and has the ability to simultaneously preserve edges on one side and suppress noise on the other side. Additionally, instead of locally summing focus values, PCA is proposed to obtain the axis of maximum variance in the filtered data.

3.2 Bilateral Filtering

An image sequence $I(X,Y,Z)$ consisting of Z images of an object, each of $X \times Y$ dimensions, is obtained by moving object away from the camera in small steps in the optical axis direction. We set i=1, 2, 3,…,X, j=1, 2, 3,…,Y and k=1, 2, 3,…,Z, indices for X, Y, and Z. By considering a small window of size $N \times N$ around a pixel, the neighborhood is arranged into a vector having dimensions $N^2 \times 1$. By collecting all these vectors along the optical axis, we obtain a matrix of size $Z \times n$ ($n = N^2$). We then apply bilateral filter on this matrix.

Bilateral filtering is a non-linear filtering approach [15] and is effective for noise removal from images while preserving the edges. Both the intensities and locations of a pixel's neighborhood are weighted through Gaussian function. In other words, it is a combination of two Gaussian filters; one concerning pixel's intensity and the other relating to pixel's spatial position. A small

neighborhood (x, y) around central pixel (x_0, y_0) is considered and formulae (1)-(2) are used to compute weight matrices $r(x, y)$ and $g(x, y)$ for the range and domain filters, respectively.

$$r(x, y) = \exp\left[- \frac{\left(f(x, y) - f(x_0, y_0)\right)^2}{2\sigma_r^2} \right] \tag{1}$$

$$g(x, y) = \exp\left[- \frac{x^2 + y^2}{2\sigma_d} \right] \tag{2}$$

The response of BF denoted by $h(x_0, y_0)$ is then computed by aggregating the product of original pixel intensities $f(x, y)$ and the output from (1) and (2).

$$h(x_0, y_0) = \frac{1}{P} \sum_{i=1}^{q} f(x, y) \times r(x, y) \times g(x, y) \tag{3}$$

where P is a normalizing factor and q is the neighborhood size. Optimal values for the parameters σ_r and σ_d in equation (1) and (2) are obtained by using Genetic Algorithms (GA). A bit string for both parameters is constructed as a chromosome. An initial population consisting of 30 chromosomes is considered. Minimization of the squared difference between computed and the actual depth value is used as fitness criteria. The algorithm is cycled for 50 generations and the optimal values of the parameters are computed.

3.3 Transformation into Eigenspace

PCA has been widely used in feature extraction and pattern recognition applications [16]. After applying BF, we obtain a resultant matrix $\mathbf{M}$ of the size $Z \times n$. PCA is applied on this matrix $\mathbf{M} = [\mathbf{m}_{kn}]$. Mean vector μ_n and covariance matrix $\mathbf{C}$ are computed by using formulae (4) and (5) respectively.

$$\boldsymbol{\mu}_n = \frac{1}{Z} \sum_{k=1}^{Z} \mathbf{m}_{kn} \tag{4}$$

$$\mathbf{C} = \frac{1}{Z-1} \sum_{k=1}^{Z} (\mathbf{m}_{kn} - \boldsymbol{\mu}_n)(\mathbf{m}_{kn} - \boldsymbol{\mu}_n)^T \tag{5}$$

Eigenvalues and their corresponding eigenvectors are computed for the matrix $\mathbf{C}$. If columns of $\mathbf{E}$ are the set of ordered eigenvectors, then the transformed data $\mathbf{F}$ in eigenspace is obtained by multiplying the mean subtracted data with the matrix $\mathbf{E}$. The columns of the transformed matrix F are known as the principal components or features in eigenspace.

$$\mathbf{F} = \mathbf{E}(\mathbf{m}_{kn} - \boldsymbol{\mu}_n) \tag{6}$$

3.4 Depth Map Generation

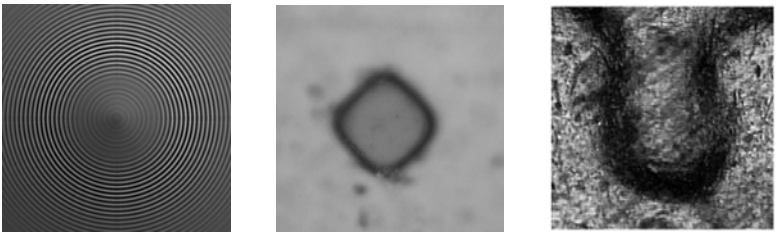
The analysis of the features obtained by PCA shows that the first principal component provides high discriminating power regarding focus information as compared to others. Therefore, it is reasonable to compute the depth map using only first feature (discussed in section 4.2). The depth value for an object point (i, j) is computed by using formula (7).

$$\mathbf{D}_{(i,j)} = \max_{k} \arg\left|\mathbf{f}_{k1}\right| \quad , \quad 1 \le k \le Z \tag{7}$$

It locates the position of the maximum absolute value in the first component $\mathbf{f}_{k1}$ .The algorithm iterates *XY* times to compute the complete depth map for the object.

4 Results and Discussion

Image sequences of synthetic and real objects were experimented for evaluating the performance of the proposed method. A sequence of 97 images of simulated cone was generated synthetically by using simulation software. Two image sequences of real objects TFT-LCD color filter and coin, consisting each of 60 images, were obtained from the microscope control system (MCS). The details about MCS can be found in [5]. Sample images of these objects are shown in Fig. 1.



(a) (b) (c)

Fig. 1 Images of sample objects (a) Simulated cone (b) TFT-LCD color filter (c) coin

4.1 Performance Measures

The performance of a focus measure is usually gauged on the basis of unimodality and monotonicity of the focus curve. Yang et al. [17] used the resolution metric for the focus measure evaluation. It computes the deviation of the fitted (Gaussian or quadratic) focus curve from the best focused position. According to the Heisenberg uncertainty principle, the resolution *R* for a focus measure function is defined as

$$R = \frac{1}{\|f\|^2} \sum_{k=1}^{Z} (z_k - z_o)^2 (f(z_k))^2 \tag{8}$$

where z_o is the lens position corresponding to the best focus point, Z is the number of total lens positions and $\|f\|$ is the Euclidian norm of the focus measure function. The sharper curve near the best focus position indicates better results and depicting that how well the out-of-focus features are suppressed. In addition, two global statistical metrics, Root Mean Square Error (RMSE) and correlation have also been applied for the evaluation of depth maps using synthetic image sequence. The RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{1}{XY} \sum_{x=1}^{X} \sum_{y=1}^{Y} |f(x,y) - g(x,y)|^2} \tag{9}$$

where X is number of pixels in horizontal direction, Y is number of vertical pixels in the image, $f(x, y)$ is the original image, and $g(x, y)$ is the processed image. The smaller the RMSE, the better the result will be. The correlation metric provides the similarity between the two depth maps. The higher the correlation, the results will be better.

4.2 Analysis in Eigenspace

Fig. 2 shows the different intermediate steps of the proposed algorithm. The object point (200,200) of the simulated cone is considered for experimentation. Fig. 2(a) represents the original pixels intensities for 97 positions by considering nine pixels for each position (3×3 window size). While fig. 2(b) shows the corresponding resultants values obtained through bilateral filtering. After applying PCA, the data is shown in eigenspace through Fig. 2(c). It can be observed that the first component in eigenspace provides a sharper and narrower peak among others. Lower order components contain noisy information regarding focus position and thus can be discarded.

4.3 Accuracy Comparison

The performance of the proposed algorithm is compared with SML, TEN, GLV, and DCT1 (ratio of AC and DC energies) using resolution, RMSE, and correlation metrics. Table 1 compares the resolution values computed for three object points from each test object, simulated cone, TFT-LCD color filter, and coin. We fitted the obtained focus curves to the Gaussian model and rescaled the function values to (0, 1). The resolution was computed by using seven positions on each side of the best focus position (peak). For the point (200,200) of simulated cone, the 75th position out of total 97 steps is considered to be the best focused position. The resolution computed for the proposed method was 1.9173, which was the smallest

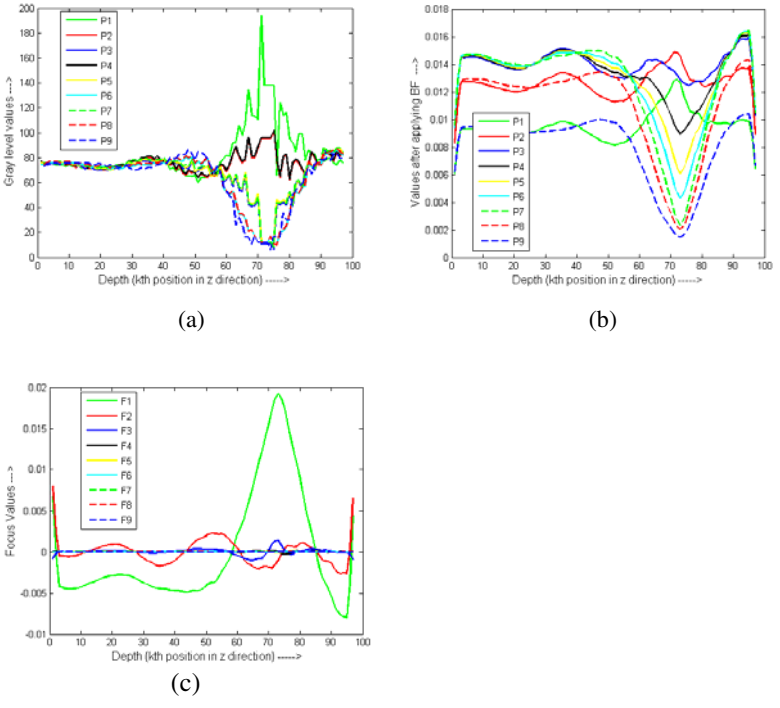


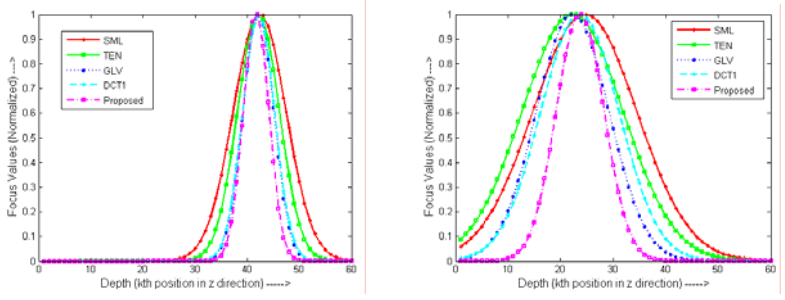
Fig. 2 Algorithm Analysis (a) Original data (b) Data obtained after applying bilateral filtering (c) Transformed data into eigenspace

among all others. The resolution values of the object points (140,140) for the TFT-LCD color filter and (245,245) for the coin, having the best-focused positions at 43rd and 23rd frames respectively, were also computed. Table 1 compares the computed resolution values for different methods. It can be concluded that the proposed method has provided the lowest resolution values as compared with other methods.

Table 1 Resolution Comparisons

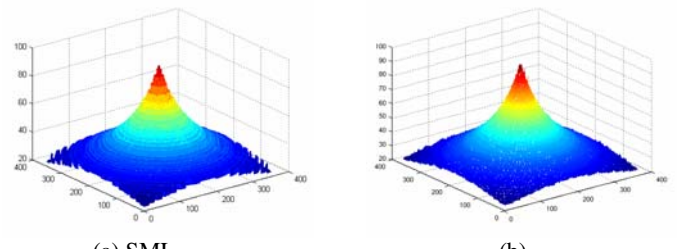
Algorithm	Simulated cone	TFT-LCD	Coin
SML	5.5603	3.8631	7.1269
TEN	4.6644	2.2308	5.5956
GLV	2.4331	0.7105	4.2162
DCT1	3.3627	0.8557	5.8686
Proposed	1.9173	0.1068	3.8336

Table-2 shows the comparison of the RMSE and the correlation values obtained from different SFF methods using the image sequence of the simulated cone. The results obtained from the proposed method are comparable with original depth map of the simulated cone, while the RMSE value is the lowest one among others.

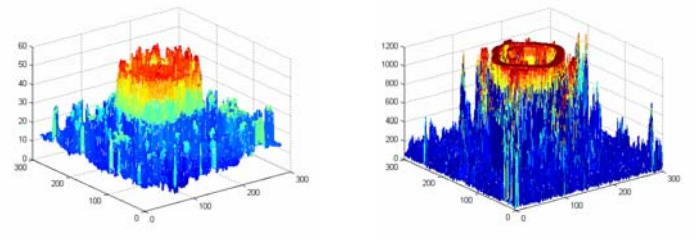


(a) (b)

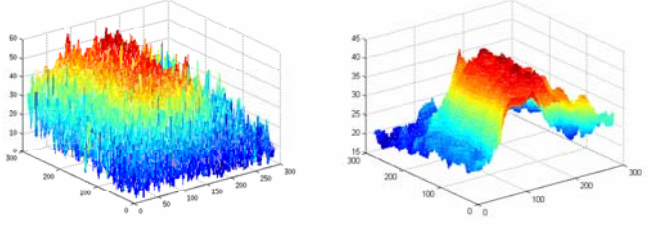
Fig. 3 Fitted focus measure curves to Gaussian model for (a) TFT-LCD color filter object point (140,140) (b) coin object point (245,245)



(a) SML (b)



(c) GLV (d)



(e) DCT1 (f)

Fig. 4 3D shapes reconstructed (a-b) simulated cone (c-d) TFT-LCD color filter (e-f) coin. (right column) 3D shapes recovered by proposed method

Table 2 Performance Comparison

SFF Method	RMSE	Correlation
SML	8.7383	0.8920
TEN	8.7872	0.8949
GLV	9.3615	0.8905
DCT1	8.6970	0.9170
Proposed	8.3521	0.9453

Fig.3 shows the focus curves fitted to the Gaussian model for the object points (140,140), (245,245) of TFT-LCD color filter and coin, respectively. We can observe that the curves obtained by proposed method are narrower and sharper near the best-focused positions as compared to existing methods.

Fig. 4 shows the 3D shapes reconstructed for the test objects. It can be seen from the right column of the Fig. 4 that the 3D shapes recovered by the proposed approach are smoother and finer than those recovered by using other methods. It is also notable that the shapes reconstructed by existing approaches have coarse surface due to the local summation of focus values.

5 Conclusion

In this paper, we introduced a new method for SFF based on BF and PCA. Firstly, the sequence of neighborhood is convolved with BF and then in the second step, PCA is applied on the resultant matrix to transform data into eigenspace. By using BF, we achieved robustness against noise while preserving edges. On the other hand, by applying PCA, we are able to obtain the axis of maximum variance in the data. The position of the maximum absolute value from the score of the first component in eigenspace has been taken as depth value. The experimental results have shown the effectiveness of the proposed method. The research could further be extended to investigate the employment of intelligent approaches [18]-[20] for proper feature extraction and their subsequent exploitation for accurate depth map generation.

References

1. Bilal, A.M., Choi, T.S.: A Heuristic approach for finding best focused shape. IEEE Transactions on Circuits and Systems for Video Technology 15(4), 566–574 (2005)
2. Malik, A.S., Choi, T.S.: Application of Passive Techniques for Three Dimensional Cameras. IEEE Trans. Consumer Electronics 53(2) (2007)
3. Krotkov, E.: Focusing. Int. J. Computer Vision 1, 223–237 (1987)
4. Malik, A.S., Choi, T.S.: Consideration of illumination effects and optimization of window size for accurate calculation of depth map for 3D shape recovery. Pattern Recognition 40(1), 154–170 (2007)
5. Bilal, A.M., Choi, T.S.: Application of Three Dimensional Shape from Image Focus in LCD/TFT Displays Manufacturing. IEEE Trans. Consumer Electronics 53(1), 1–4 (2007)

6. Sun, Y., Nelson, D., Bradley, J.: Autofocusing in computer microscopy: Selecting the optimal focus algorithm. Microscopy Research and Technique 65(3), 139–149 (2004)
7. Nayar, S.K., Nakagawa, Y.: Shape from focus. IEEE Trans. Pattern Anal. Mach. Intell. 16(8), 824–831 (1994)
8. Tenenbaum, J.M.: Accommodation in computer vision. Ph.D. Thesis, Stanford Univ. (1970)
9. Baina, J., Dublet, J.: Automatic focus and iris control for video cameras. In: IEEE Fifth International Conference on Image Processing and its Applications, pp. 232–235 (1995)
10. Shen, C.-H., Chen, H.H.: Robust focus measure for low-contrast images. In: Int. Conference on Consumer Electronics, ICCE 2006. Digest of Technical Papers, pp. 69–70 (2006)
11. Kristan, M., Perse, J., Perse, M., Kovacic, S.: A Bayes-spectral-entropy-based measure of camera focus using a discrete cosine transform. Pattern Recognition Letters 27(13), 1431–1439 (2006)
12. Subbarao, M., Choi, T.S.: Accurate recovery of three dimensional shape from image focus. IEEE Trans. Pattern Anal. Mach. Intell. 17(3), 266–274 (1995)
13. Yun, J., Choi, T.S.: Accurate 3-D shape recovery using curved window focus measure. In: ICIP 1999, vol. 3, pp. 910–914 (1999)
14. Mahmood, M.T., Choi, T.S.: A feature analysis approach for 3D shape recovery from image focus. In: 15th IEEE International Conference on Image Processing, pp. 3216–3219 (2008)
15. Tomasi, C., Manduchi, R.: Bilateral Filtering for gray and color images. In: Sixth Int. Conf. on Computer Vision, New Delhi, India, pp. 839–846 (1998)
16. Ravi, V., Pramodh, C.: Threshold accepting trained principal component neural network and feature subset selection: Application to bankruptcy prediction in banks. Applied Soft Computing 8(4), 1539–1548 (2008)
17. Yang, G., Nelson, B.J.: Wavelet-based autofocusing and unsupervised segmentation of microscopic images. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, vol. 3, pp. 2143–2148 (2003)
18. Khan, A., Tahir, S.F., Majid, A., Choi, T.-S.: Machine Learning based Adaptive Watermark Decoding in View of an Anticipated Attack. Pattern Recognition 41, 2594–2610 (2008)
19. Huang, C.-L., Dun, J.-F.: A distributed PSO–SVM hybrid system with feature selection and parameter optimization. Applied Soft Computing 8(4), 1381–1391 (2008)
20. Larranãga, P., Moral, S.: Probabilistic graphical models in artificial intelligence. Appl. Soft Comput. J. (2008), doi:10.1016/j.asoc.2008.01.003

Detecting Hidden Information from Watermarked Signal Using Granulation Based Fitness Approximation

Mohsen Davarynejad, Saeed Sedghi, Majid Bahrepour, Chang Wook Ahn, Mohammad-Reza Akbarzadeht, and Carlos Artemio Coello Coello

Abstract. Spread spectrum audio watermarking (SSW) is one of the most secure techniques of audio watermarking. SSW hides information by spreading their spectrum which is called watermark and adds it to a host signal as a watermarked signal. Spreading spectrum is done by a pseudo-noise (PN) sequence. In conventional SSW approaches, the receiver must know the PN sequence used at the transmitter as well as the location of the watermark in watermarked signal for detecting hidden information. This method is attributed high security features, since any unauthorized user who does not access this information cannot detect any hidden information. Detection of the PN sequence is the key factor for detection of hidden information from SSW. Although PN sequence detection is possible by using heuristic approaches such as evolutionary algorithms, due to the high computational cost of this task, such heuristic tends to become too expensive (computationally speaking), which can turn it impractical. Much of the computational complexity involved in the use of evolutionary algorithms as an optimization tool is due to the fitness function evaluation that may either be very difficult to define or be computationally very expensive. This paper proposes the use of fitness granulation to recover a PN sequence with a chip period equal to 63, 127, 255 bits. This is a new application of authors' earlier work on adaptive fitness function approximation with fuzzy supervisory. With the proposed approach, the expensive fitness evaluation step is

Mohsen Davarynejad
Faculty of Technology, Policy and Management, Delft University of Technology
e-mail: M.Davarynejad@tudelft.nl

Saeed Sedghi · Majid Bahrepour
Department of Computer Science, University of Twente
e-mail: s.sedghi@utwente.nl, m.bahrepour@utwente.nl

Chang Wook Ahn
School of Information and Communication Engineering, Sungkyunkwan University
e-mail: cwan@skku.edu

Mohammad-Reza Akbarzadeht
Electrical Engineering Department, Ferdowsi University of Mashhad
e-mail: akbarzadeh@kiaeee.org

Carlos Artemio Coello Coello
CINVESTAV-IPN (Evolutionary Computation Group), Departmento de Computación,
Av. IPN No. 2508, Col. San Pedro Zacatenco, Mexico D.F. 07300
e-mail: ccoello@cs.cinvestav.mx

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 463 – 472.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

replaced by an approximate model. The approach is then compared with standard application of evolutionary algorithms; statistical analysis confirms that the proposed approach demonstrates an ability to reduce the computational complexity of the design problem without sacrificing performance.

1 Introduction

In recent years, digital watermarking has received considerable attention from the security and cryptographic research communities. Digital watermarking is a technique for hiding information bits into an innocuous-looking media object, which is called host, so that no one can suspect of the existence of hidden information. It is intended to provide a degree of copyright protection such as the use of digital media mushrooms [1]. Depending on the type of the host signal used to cover the hidden information, watermarking is classified into image watermarking and audio watermarking. In this paper we focus our results to audio watermarking but the approach is also applicable to image watermarking.

Numerous audio watermarking techniques have been proposed, the most important being: LSB [2], Phase coding [3], Echo hiding [4] and spread spectrum watermarking (SSW) [5]. The latter, SSW, is recognized as the most promising watermarking method because of its high robustness against noise and its high perceptual transparency. The main idea of SSW is adding spread spectrum of hidden information to the spectrum of the host signal. Spreading the spectrum of hidden information is performed by a pseudo-random noise sequence.

The detection of hidden information from the received watermark signal is performed by the exact PN sequence used for spreading the spectrum of hidden information. Therefore, the receiver should access the PN sequence for detection. This essential knowledge for detection, results in a highly secure transmission of information against any unauthorized user who does not have access to the PN sequence and location of the watermark. Hence, the PN sequence can be regarded as a secret key which is shared between the transmitter and the receiver.

In [6], a GA is presented in such a way that it is possible to detect hidden information, whereas the receiver has no knowledge of the transmitter's spreading sequence. Repeated fitness function evaluations for such a complex problem is often the most prohibitive and limiting feature of this approach. For the problem of recovering the PN sequence, sequences with different periods have different converging times. The studies have shown that the converging time increases exponentially as the period of the PN sequences increases [6]. So the approach fails by losing the validity of information. To alleviate this problem, a variety of techniques for constructing approximation models– often referred to as metamodels – have been proposed. For computationally expensive optimization problems such as detection of hidden information, it may be necessary to perform an exact evaluation and then use an approximate fitness model that is computationally efficient.

A popular subclass of fitness function approximation methods is fitness inheritance where fitness is simply transmitted (or "inherited") [7, 8]. A similar approach named "Fast Evaluation Strategy" (FES) has also been suggested in [9] for fitness approximation where the fitness of a child individual is the weighted

sum of its parents. Other common approaches based on learning and interpolation from known fitness values of a small population, (e.g. low-order polynomials and the least square estimations [10], artificial neural networks (ANN), including multi-layer perceptrons [11] and radial basis function networks [12], support vector machine (SVM) [13], etc.) have also been employed.

In this paper, the concept of fitness granulation is applied to exploit the natural tolerance of evolutionary algorithms in fitness function computations. Nature's "survival of the fittest" principle is not about exact measures of fitness; instead, it is about rankings among competing peers. By exploiting this natural tolerance for imprecision, optimization performance can be preserved by computing fitness only selectively and by preserving this ranking (based on fitness values) among individuals in a given population. Also, fitness is not interpolated or estimated; rather, the similarity and indistinguishability among real solutions is exploited.

In the proposed algorithm, as explained in detail by its authors in [14, 15], an adaptive pool of solutions (fuzzy granules) with an exactly computed fitness function is maintained. If a new individual is sufficiently similar to a known fuzzy granule, then that granule's fitness is used instead as a crude estimate. Otherwise, that individual is added to the pool as a new fuzzy granule. In this fashion, regardless of the competition's outcome, fitness of the new individual is always a physically realizable one, even if it is a "crude" estimate and not an exact measurement. The pool size as well as each of the granule's radius of influence is adaptive and will grow/shrink depending on the utility of each granule and the overall population fitness. To encourage fewer function evaluations, each granule's radius of influence is initially large and is gradually shrunk at later stages of evolution. This encourages more exact fitness evaluations when competition is fierce among more similar and converging solutions. Furthermore, to prevent the pool from growing too large, granules that are not used are gradually eliminated. This fuzzy granulation scheme is applied here as a type of fuzzy approximation model.

The paper is organised as follows: Section 2 presents a brief overview of the proposed granulation based fitness approximation method. For future details, readers are referred to [15] were the proposed method is described in more details and an example is also provided in addition to some supporting simulation. An auto-tuning strategy for determining the width of membership functions (MFs) is presented in [16], which removes the need for exact parameter determination, without obvious influence on convergence speed. In Section 3, the spread spectrum watermarking and the properties of the PN sequence are described. In Section 4, the recovering of the PN sequence from a received watermarked signal using GA and granulation based fitness approximation is presented. Some supporting simulation results and discussion thereof are presented in Section 5. Finally, some conclusions are drawn in Section 6.

2 AFFG Framework - The Main Idea

The proposed adaptive fuzzy fitness granulation (AFFG) aims to minimize the number of exact fitness function evaluations by creating a pool of solutions (fuzzy

granules) by which an approximate solution may be sufficiently applied to proceed with the evolution. The algorithm uses Fuzzy Similarity Analysis (FSA) to produce and update an adaptive competitive pool of dissimilar solutions/granules. When a new solution is introduced to this pool, granules compete by a measure of similarity to win the new solution and thereby to prolong their lives in the pool. In turn, the new individual simply assumes fitness of the winning (most similar) individual in this pool. If none of the granules is sufficiently similar to the new individual, i.e. their similarity is below a certain threshold, the new individual is instead added to the pool after its fitness is evaluated exactly by the known fitness function. Finally, granules that cannot win new individuals are gradually eliminated in order to avoid a continuously enlarging pool. The proposed algorithm is briefly discussed below. For further details, readers are referred to [14, 15] where the proposed method is described in more details and an example is provided, in addition to some supporting simulation.

After a random parent population is initially created, a set of fuzzy granules that is initially empty is shaped. The average similarity of new solutions to each granule is then computed. This is influenced by granule enlargement/shrinkage. The fitness of each new solution is either calculated by exact fitness function computing or is estimated by associating it to one of the granules in the pool if there is a granule in the pool with a similarity value higher than a predefined threshold. Depending on the complexity of the problem, the size of this pool can become excessive and computationally cumbersome by itself. To prevent such unnecessary computational effort, an interesting and advantageous approach is introduced in [15].

The distance measurement parameter is completely influenced by granule enlargement/shrinkage in widths of the produced MFs. In [12], the combined effect of granule enlargement/shrinkage is based on the granule fitness and it needs to adjust two parameters. These parameters are problem dependent and it seems critical to set up a procedure in order to avoid this difficulty. In order to removes the need for exact parameter determination of AFFG approach, an auto-tuning strategy is presented in [16].

3 Spread Spectrum Watermarking

SSW is the algorithm that has high robustness (surviving hidden information after noise addition), high transparency (high quality of watermarked signals after addition of hidden information) and high security (against unauthorized users) from the watermarking features point if view. SSW borrows the idea of spread spectrum communication to hide information by embedding the bits of information into a host signal. The embedding processes employ a pseudorandom noise (PN) sequence. A PN sequence is a zero mean periodic binary sequence with a noise defined by a waveform whose bits are equal to +1 or –1 [17].

Each bit of hidden information w(i) is multiplied by all the bits of a period of a pseudorandom noise (PN), p(n) sequence to generate each block of the watermark signal w(i).

$$w(i) = p(n)\ m(i) \tag{1}$$

A watermark signal is the sequence of all the watermark blocks as w = (w1,…,wk). The watermarked signal s(w,x) is produced as:

$$S(w,\ x) = \lambda\ w(n) + x(n) \tag{2}$$

Then, the watermarked signal s(w,x) is sent to the receiver.

The extraction of hidden information from a received watermarked signal is performed using the correlation property of the PN sequence. Cross correlation C(.,.) between two PN sequences pa and pb is as (3) [18]:

$$C(P_a,\ P_b) = \frac{1}{N}\sum_{i=0}^{N-1}(P_a(i)\ P_b(i)) = \begin{Bmatrix} i, & if\ a=b \\ -1/N, & otherwise \end{Bmatrix} \tag{3}$$

Hence, cross correlation between a watermarked signal and a PN sequence is as follows:

$$C(S,p') = C(w,p') + C(m.p,p') = \begin{Bmatrix} C(w,p') + m & if\ p=p' \\ C(w,p') - \frac{m}{N} & Otherwise \end{Bmatrix} \tag{4}$$

Equation (4) expresses that by calculating the correlations between the received watermarked signal and the employed PN sequence at the transmitter, and comparing the result with a threshold, determines the bit of hidden information.

4 Recovering PN Sequence

Recovering the PN sequence from a spread spectrum watermarked signal where no information about the PN sequence or its location is known would be very hard since there are vast regions for the solutions set. For instance, in order to recover a PN sequence with a period equal to 63 bits, 263 PN sequences must be generated.

In this section, with the assumption that knowing the exact location of the watermark in the watermarked signal, we describe the recovering of the PN sequence.

In [19], an approach for detecting the hidden information from an image spread spectrum signal is proposed. This approach detects abrupt jumps in the statistics of watermarked signal to recover the PN sequence. The proposed algorithm, which is based on hypothesis tests for detection of abrupt jumps in the statistical values is very complicated and its performance suffers from for low frequency embedding.

Recovering the PN sequence could be considered as an unconstrained optimization problem. We have a set of feasible solutions to minimize a cost function using a global optimizer. The set of feasible solutions are sequences with

periods of PN sequences and elements of +1 and –1. Defining a cost function for this problem should be based on a very useful property of SSW in detection which is the correlation property of the PN sequence. So, our cost function is the cross correlation between the generated sequence and the watermarked signal.

In [20], an interesting method for recovering the PN sequence of a spread spectrum signal with a predefined SNR, is proposed. This approach uses a GA with a fitness function defined in terms of the cross correlation between the estimated PN sequence and the spread spectrum. However, the spread spectrum watermarking is more complicated than a single spread spectrum signal since in SSW, the spread spectrum hidden information is like a white Gaussian noise for a host signal.

Note that computing the cross correlation between sequences of our solutions set and watermarked signal for only one block of SSW signals, will not converge to the PN sequence used at the transmitter, since the energy of the host signal is at least 12 dB more than the energy of the watermark and it has a strong effect on maximizing the cross correlation (i.e., an optimization algorithm converges to a sequence that maximizes the correlation with the host). As a solution to this problem, several consequence blocks of watermark (i.e. several bits of hidden information) should be considered in the computation of the cross correlation. In this case, the watermark signal has a stronger effect than the host signal on maximizing the cross correlation function.

Finding the global optimization by searching over the entire set of our solutions, as mentioned above, is the subject of deterministic methods such as covering methods, tunneling methods, zooming methods, etc. These methods find the global minimum by an exhaustive search over the entire solutions set. For instance, the basic idea for the covering method is to cover the feasible solutions set by evaluating the objective function at all points [21]. These algorithms have high reliability and accuracy is always guaranteed, but they have a slower convergence rate [22].

Since our solutions set is vast, we need efficient optimization algorithms that have a high reliability and fast convergence rate. Many stochastic optimization algorithms have been proposed such as the genetic algorithm, simulated annealing, ant colony, etc.; however the GA has shown to be one of the most successful and powerful search engines for a wide range of applications and strikes an attractive balance between reliability and convergence rate.

5 Empirical Results

The empirical study consisted on comparing the GA performance, as a function optimizer, and the proposed granulation techniques with fuzzy supervisory (AFFG-FS). Since in [16], by numerous simulations, it has been shown that AFFG-FS, with its fuzzy supervisory technique, removes the need for exact parameter determination without obvious influence on convergence speed, we did not take into account the original AFFG.

Since the GA was used as a function optimizer, we chose roulette wheel with elitism as the selection method, in order to keep track of the best solution found. The GA was implemented with one-point crossover. The population size was set to 20 with the elite size of 2. The mutation and crossover rate used was 0.01 and 1.0, respectively. Ten runs of each experiment were executed.

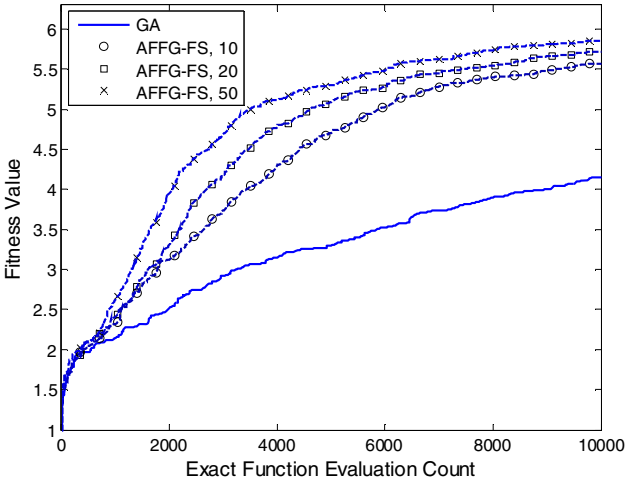


Fig. 1 Cross correlation between estimated PN sequence with the period of 255 chips and the watermarked signal

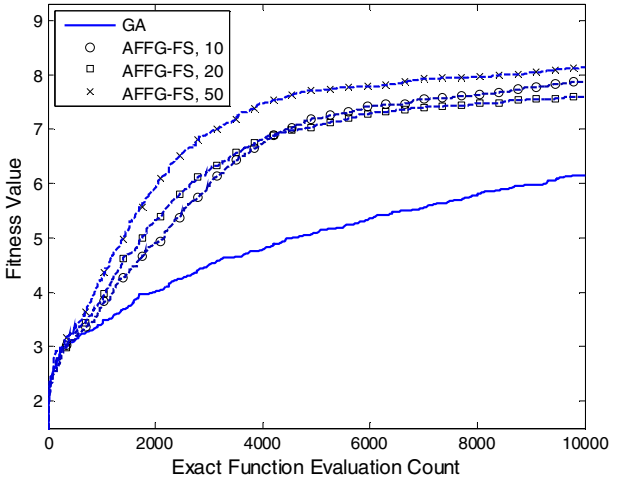


Fig. 2 Cross correlation between estimated PN sequence with the period of 127 chips and the watermarked signal

For AFFG-FS, the number of individuals in the granule pool is varied between 10, 20 and 50. The report results were obtained by achieving the same level of fitness evaluation for both the canonical GA and the proposed methods.

The average convergence trends of the standard GA, AFFG-FS are summarized in Figures 1-3. All the results presented were averaged over 10 runs. As shown in the Figures, the search performance of the AFFG-FS is superior to the standard GA even with a small number of individuals in the granule pool.

We also studied the effect of varying the number of granules N_G on the convergence behavior of AFFG-FS. It can be shown that AFFG-FS is not significantly sensitive to N_G . However, a further increase of N_G , slows down the rate of convergence due to the imposed computational complexity.

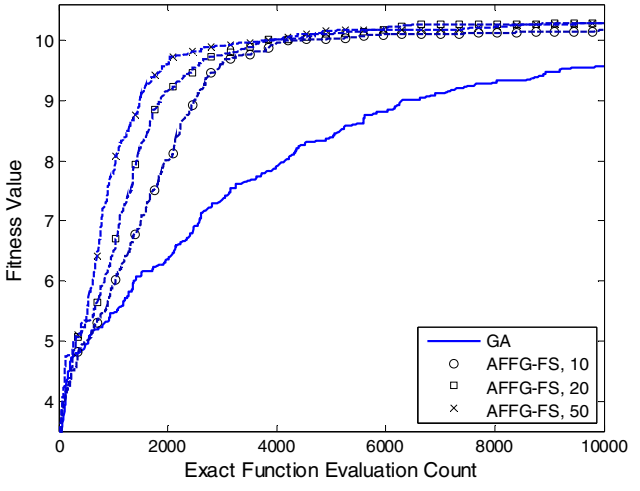


Fig. 3 Cross correlation between estimated PN sequence with the period of 63 chips and the watermarked signal

6 Concluding Remarks

One of the most secure techniques of audio watermarking is spread spectrum audio watermarking. The key factor for detection of hidden information from SSW is the PN sequence. Here, an intelligent guided technique via an adaptive fuzzy similarity analysis is adopted in-order to accelerate the process of evolutionary based recovering of PN sequence. A fuzzy supervisor such as the auto-tuning algorithm is introduced in order to avoid the tuning of parameters used in this approach.

A comparison is provided between simple GA, FES and the proposed approach. Numerical results showed that the proposed technique is capable of optimizing functions of varied complexity efficiently. Furthermore, in comparison with our previous work, it can be shown that AFFG and AFFG-FS are not significantly

sensitive to N_G, and small N_G values can still produce good results. Moreover, the auto-tuning of the fuzzy supervisor removes the need for exact parameter determination without an obvious influence on convergence speed.

Acknowledgments

The last author acknowledges support from CONACyT project no. 45683-Y.

References

1. Cvejic, N., Seppanen, T.: Algorithms for Audio Watermarking and Steganography. PHD thesis, oulu university of technology (June 2004)
2. Gopulan, K.: Audio steganography using bit modification. In: Proceedings of the 2003 International conference on Acoustic Speech and signal Processing (2003)
3. Ansari, R., Malik, H., Khikhar, A.: Data-hiding in audio using frequency-selective phase alteration. In: Proceedings of the IEEE International conference on Acoustic Speech and signal Processing (2004)
4. Joong, H., Choi, Y.H.: A novel echo-hiding scheme with forward backward kernels. IEEE Transactions on Circuits and Systems for Video Technology 13(8) (August 2003)
5. Liu, Z., Inue, A.: Spread spectrum watermarking of audio signals. IEEE Transactions on Circuits and System for Video Technology 13(8) (August 2003)
6. Sedghi, S., Mashhadi, H.R., Khademi, M.: Detecting Hidden Information from a Spread Spectrum Watermarked Signal by Genetic Algorithm. In: IEEE Congress on Evolutionary Computation, July 16-21, pp. 480–485 (2006)
7. Chen, J.-H., Goldberg, D., Ho, S.-Y., Sastry, K.: Fitness inheritance in multiobjective optimization. In: Proceedings of the 2002 International conference on Genetic and Evolutionary Computation Conference, pp. 319–326 (2002)
8. Reyes-Sierra, M., Coello, C.A.C.: Dynamic fitness inheritance proportion for multi-objective particle swarm optimization. In: Proceedings of the 8th annual conference on Genetic and Evolutionary Computation, Seattle, Washington, USA, July 8-12 (2006)
9. Salami, M., Hendtlass, T.: The Fast Evaluation Strategy for Evolvable Hardware. Genetic Programming and Evolvable Machines 6(2), 139–162 (2005)
10. Myers, R., Montgomery, D.: Response Surface Methodology. John Wiley & Sons, Inc., New York (1995)
11. Hong, Y.-S., Lee, H., Tahk, M.-J.: Acceleration of the convergence speed of evolutionary algorithms using multi-layer neural networks. Journal of Engineering Optimization 35(1), 91–102 (2003)
12. Won, K.S., Ray, T.: A Framework for Design Optimization using Surrogates. Journal of Engineering Optimization, 685–703 (2005)
13. Gunn, S.R.: Support Vector Machines for Classification and Regression, Technical Report, School of Electronics and Computer Science, University of Southampton, Southampton, U.K. (1998)
14. Davarynejad, M.: Fuzzy Fitness Granulation in Evolutionary Algorithms for Complex Optimization, M.Sc. Thesis. Ferdowsi University of Mashhad, Department of Electrical Engineering (2007)

15. Davarynejad, M., Akbarzadeh-T, M.-R., Pariz, N.: A Novel General Framework for Evolutionary Optimization: Adaptive Fuzzy Fitness Granulation. In: Proceedings of the 2007 IEEE International Conference on Evolutionary Computing, Singapore, September 25-28, pp. 951–956 (2007)
16. Davarynejad, M., Akbarzadeh-T, M.-R., Coello, C.A.C.: Auto-Tuning Fuzzy Granulation for Evolutionary Optimization. In: IEEE World Congress on Evolutionary Computation, June 2008, pp. 3572–3579 (2008)
17. Haykin, S.: Communication Systems, 4th edn. John Wiley & Sons, Inc., Chichester (2001)
18. Liu, Z., Kobayashi, Y., Sawato, S., Inoue, A.: A robust audio watermarking method using sine function patterns based on pseudorandom sequences. In: Proceedings of the Pacific Rim Workshop on Digital Steganography (2002)
19. Trivedi, S., Chandramouli, R.: Secret Key Estimation in Sequential Steganography. IEEE Transaction on Signal Processing 53(2) (February 2005)
20. Asghari, V.R., Ardebilipour, M.: Spread Spectrum Code Estimation by Genetic Algorithm. International Journal of Signal Processing 1(4) (2004)
21. Arora, J.S., Elwakeil, O.A., Chahande, A.: Global optimization methods for engeering applications: a review. Optimal Design Laboratory (1995)
22. Yen, K., Hanzo, L.: Genetic Algorithm Assisted Joint Multiuser Symbol Detection and Fading Channel Estimation for Synchronous CDMA Systems. IEEE Transaction on selected areas in communication 19(6) (June 2001)

Fuzzy Approaches for Colour Image Palette Selection

Gerald Schaefer and Huiyu Zhou

Abstract. Colour quantisation algorithms are used to display true colour images using a limited palette of distinct colours. The choice of a good colour palette is crucial as it directly determines the quality of the resulting image. Colour quantisation can also be seen as a clustering problem where the task is to identify those clusters that best represent the colours in an image. In this paper we investigate the performance of various fuzzy c-means clustering algorithms for colour quantisation of images. In particular, we use conventional fuzzy c-means as well as some more efficient variants thereof, namely fast fuzzy c-means with random sampling, fast generalised fuzzy c-means, and anisotropic mean shift based fuzzy c-means algorithm. Experimental results show that fuzzy c-means performs significantly better than other, purpose built colour quantisation algorithms, and also confirm that the fast fuzzy clustering algorithms provide quantisation results similar to the full conventional fuzzy c-means approach.

1 Introduction

Colour quantisation is a common image processing technique that allows the representation of true colour images using only a small number of colours. True colour images typically use 24 bits per pixel resulting overall in 2^{24}, i.e. more than 16 million different colours. Colour quantisation uses a colour palette that contains only a small number of distinct colours (usually between 8 and 256) and pixel data are then stored as indices to this palette. Clearly the choice of the colours that make up

Gerald Schaefer
School of Engineering and Applied Science, Aston University, Birmingham, U.K.
e-mail: G.Schaefer@aston.ac.uk

Huiyu Zhou
School of Engineering and Design, Brunel University, Uxbridge, U.K.
e-mail: Huiyu.Zhou@brunel.ac.uk

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 473–482.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

the palette is of crucial importance for the quality of the quantised image. However, the selection of the optimal colour palette is known to be an np-hard problem [10]. In the image processing literature many different algorithms have been introduced that aim to find a palette that allows for good image quality of the quantised image [10, 9, 8]. Soft computing techniques such as genetic algorithms have also been employed to extract a suitable palette [13, 12].

Colour quantisation can also be seen as a clustering problem where the task is to identify those clusters that best represent the colours in an image. In this paper we investigate the performance of various fuzzy c-means clustering algorithms for colour quantisation of images. In particular, we use conventional fuzzy c-means [2] as well as some more efficient variants thereof, namely fast fuzzy c-means with random sampling [4], fast generalised fuzzy c-means [4], and anisotropic mean shift based fuzzy c-means algorithm [17]. Experimental results show that fuzzy c-means performs significantly better than other, purpose built colour quantisation algorithms, and also confirm that the fast fuzzy clustering algorithms provide quantisation results similar to the full conventional fuzzy c-means approach.

2 Fuzzy c-Means

Fuzzy c-means (FCM) is based on the idea of finding cluster centres by iteratively adjusting their positions and evaluation of an objective function which for colour quantisation can be formulated as

$$E = \sum_{j=1}^{C} \sum_{i=1}^{N} \mu_{ij}^{k} ||x_i - c_j||^2 \tag{1}$$

where μ_{ij}^{k} is the fuzzy membership of pixel x_i and the colour cluster identified by its centre c_j, k is a constant that defines the fuzziness of the resulting partitions, and C and N are the number of clusters and total number of image pixels.

E can reach the global minimum when pixels nearby the centroid of corresponding colour clusters are assigned higher membership values, while lower membership values are assigned to pixels far from the centroid [5]. Membership is proportional to the probability that a pixel belongs to a specific cluster where the probability is only dependent on the distance between the image pixel and each independent cluster centre. Membership functions and cluster centres are updated by

$$\mu_{ij} = \frac{1}{\sum_{m=1}^{C} \left(\frac{||x_j - c_i||}{||x_j - c_m||)^{2/(k-1)}} \right)} \tag{2}$$

and

$$c_i = \frac{\sum_{j=1}^{N} \mu_{ij}^{k} x_j}{\sum_{j=1}^{N} \mu_{ij}^{k}}. \tag{3}$$

The steps involved in fuzzy c-means clustering are [2]:

1. Initialise the cluster centres c_i and let $t = 0$.
2. Initialise the fuzzy partition memberships functions μ_{ij} according to Eq. (2).
3. Let $t = t + 1$ and compute new cluster centres c_i using Eq. (3).
4. Repeat Steps 2 to 3 until convergence.

An initial setting for each cluster centre is required and FCM can be shown to converge to a local minimum.

3 Fast FCM with Random Sampling (RSFCM)

To combat the computational complexity of FCM [11], Cheng *et al.* [4] proposed a multistage random sampling strategy. This method has a lower number of feature vectors and also requires fewer iterations to converge. The basic idea is to randomly sample and obtain a small subset of the dataset in order to approximate the cluster centres of the full dataset. This approximation is then used to reduce the number of iterations. The random sampling FCM algorithm consists of two phases. First, a multistage iterative process of a modified FCM is performed. Phase 2 is then a standard FCM with the cluster centres approximated by the final cluster centres from Phase 1.

Phase 1: Randomly initialise the cluster centres c_i
Let $X_{\Delta\%}$ be a subset whose number of subsamples is $\Delta\%$ of the N samples contained in the full dataset X and denote the number of stages as n. ε_1 and ε_2 are parameters used as stopping criteria. After the following steps the dataset (denoted as $X_{(n_s * \Delta\%)}$) will include $N * \Delta\%$ samples:

1. Select $X_{(\Delta\%)}$ from the set of the original feature vectors matrix (z = 1).
2. Initialise the fuzzy memberships functions μ_{ij} using Equation (2) with $X_{(z*\Delta\%)}$.
3. Compute the stopping condition $\varepsilon = \varepsilon_1$-z$*((\varepsilon_1$-$\varepsilon_2)/n_s)$ and let $t = 0$.
4. Set $t = t + 1$.
5. Compute the cluster centres $c_{(z*\Delta\%)}$ using Equation (3).
6. Compute $\mu_{(z*\Delta\%)}$ using Equation (2).
7. If $||\mu^j_{(z*\Delta\%)} - \mu^{j-1}_{(z*\Delta\%)}|| \geq \varepsilon$, then go to Step 4.
8. If $z \leq n_s$ then select another $X_{(\Delta\%)}$ and merge it with the current $X_{(z*\Delta\%)}$ and set $z = z + 1$, otherwise move to Phase 2 of the algorithm.

Phase 2: FCM clustering
1. Initialise μ_{ij} using the results from Phase 1, i.e. $c_{(n_s*\Delta\%)}$ with Equation (3) for the full data set
2. Go to Steps 3 of the conventional FCM algorithm and iterate the algorithm stopping criterion ε_2 is met.

Evidence has shown that this improved FCM with random sampling is able to reduce the computation requested in the classical FCM method.

4 Fast Generalized FCM Scheme (EnFCM)

Ahmed *et al.* [1] introduced an alternative to the classical FCM by adding a term that enables the labelling of a pixel to be associated with its neighbourhood. As a regulator, the neighbourhood term can change the solution towards piecewise homogeneous labelling. As a further extension of this work, Szilágyi *et al.* [14] proposed their EnFCM algorithm to speed up the clustering process. In order to reduce the computational complexity, a linearity-weighted sum image g is formed from the original image, and the local neighbour average image evaluated as

$$g_m = \frac{1}{1+\alpha}\left(x_m + \frac{\alpha}{N_R}\sum_{j\in N_r} x_j\right) \tag{4}$$

where g_m denotes the pixel value of the m-th pixel of the image g, x_j represents the neighbours of x_m, N_R is the cardinality of a cluster, and N_r represents the set of neighbours falling into a window around x_m.

The objective function used is defined as

$$J = \sum_{i=1}^{C}\sum_{i=1}^{q_c} \gamma_l \mu_{ij}^m (g_l - c_i)^2 \tag{5}$$

where q_c denotes the number of colours in the image, and γ_l is the number of the pixels having colour l with $l = 1,2,\ldots,q_c$. Thus, $\sum_{l=1}^{q_c}\gamma_l = N$ under the constraint that $\sum_{i=1}^{C}\mu_{ij} = 1$ for any l.

Finally, we can obtain the following expressions for membership functions and cluster centres [3]:

$$\mu_{il} = \frac{(g_l - c_i)^{-2/m-1}}{\sum_{j=1}^{C}(g_l - c_j)^{-2/m-1}} \tag{6}$$

and

$$s_i = \frac{\sum_{l=1}^{q_c}\gamma_l \mu_{il}^m g_l}{\sum_{l=1}^{q_c}\gamma_l \mu_{il}^m}. \tag{7}$$

EnFCM considers a number of pixels of similar colour as a weight. Thus, this process may accelerate the convergence of searching for global similarity. Cai *et al.* [3] utilise a measure S_{ij}, which incorporates the local spatial relationship S_{ij}^s and the local colour relationship S_{ij}^g, and is defined as

$$S_{ij} = \begin{cases} S_{ij}^s \times S_{ij}^g, & j \neq i \\ 0, & j = i \end{cases} \tag{8}$$

with

$$S_{ij}^s = \exp\left(\frac{-\max(|p_{cj} - p_{ci}|, |q_{cj} - q_{ci}|)}{\lambda_s}\right) \tag{9}$$

and

$$S_{ij}^{g} = \exp\left(\frac{-||x_i - x_j||^2}{\lambda_g \times \sigma_g^2}\right) \tag{10}$$

where (p_{ci}, q_{ci}) describe the co-ordinates of the i-th pixel, σ_g is a global scale factor of the spread of S_{ij}^{s}, and λ_s and λ_g represent scaling factors. S_{ij} replaces α in Eq. (4).

Hence, the newly generated image g is updated as

$$g_i = \frac{\sum_{j \in N_i} S_{ij} x_j}{S_{ij}} \tag{11}$$

and is restricted to [0, 255] due to the denominator.

Given a pre-defined number of clusters C and a threshold value $\varepsilon > 0$, the fast generalised FCM algorithm proceeds in the following steps:

1. Initialise clusters c_j.
2. Compute the local similarity measures S_{ij} using Equation (8) for all neighbours and windows over the image.
3. Compute linearly-weighted summed image g using Equation (11).
4. Update the membership partitions using Equation (6).
5. Update the cluster centres c_i using Equation (7).
6. If $\sum_{i=1}^{C} ||c_{i(old)} - c_{i(new)}||^2 > \varepsilon$ go to Step 4.

5 Anisotropic Mean Shift Based FCM (AMSFCM)

Anisotropic mean shift based FCM [17] utilises an anisotropic mean shift algorithm coupled with fuzzy clustering. Mean shift based techniques have been shown to be capable of estimating the local density gradients of similar pixels. These gradient estimates are iteratively performed so that all pixels can find similar pixels in the same image [6, 7]. A standard mean shift approach method uses radially symmetric kernels. Unfortunately, the temporal coherence will be reduced in the presence of irregular structures and noise in the image. This reduced coherence may not be properly detected by radially symmetric kernels and thus, an improved mean shift approach, namely anisotropic kernel mean shift [15], provides better performance.

In mean shift algorithms the image clusters are iteratively moved along the gradient of the density function before they become stationary. Those points gathering in an outlined area are treated as the members of the same segment. A kernel density estimate is defined by

$$\tilde{f}(x) = \frac{1}{N} \sum_{i=1}^{N} K(x - x_i) \tag{12}$$

with

$$K(x) = |H|^{-0.5} K(H^{-0.5} x) \tag{13}$$

where N is the number of samples, and x_i stands for a sample from an unknown density function f. $K(\cdot)$ is the d-variate kernel function with compact support satisfying the regularity constraints, and H is a symmetric positive definite $d \times d$ bandwidth

matrix. Usually, we have $K(x) = k_e(\phi)$, where $k_e(\phi)$ is a convex decreasing function, e.g. for a Gaussian kernel

$$k_e(\phi) = c_t e^{-\phi/2} \tag{14}$$

or for an Epanechnikov kernel

$$k_e(\phi) = c_t \max(1-\phi, 0) \tag{15}$$

where c_t is a normalising constant.

If a single global spherical bandwidth is applied, $H = h^2\mathbf{I}$ (where $\mathbf{I}$ is the identity matrix), then we have

$$\tilde{f}(x) = \frac{1}{Nh^d} \sum_{i=1}^{N} K\left(\frac{x-x_i}{h}\right). \tag{16}$$

Since the kernel can be divided into two different radially symmetric kernels, we have the kernel density estimate as

$$\tilde{f}(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{h^\beta (H_i^\alpha)^q} k^\alpha \cdot (d(c_i^\alpha, x_i^\alpha, H_i^\alpha)) k^\beta \left(||(c_i^\beta - x_i^\beta)/(h^\beta (H_i^\alpha))||^2 \right) \tag{17}$$

where α and β denote the spatial and temporal components respectively and $d(c_i^\alpha, x_i^\alpha, H_i^\alpha)$ is the Mahalanobis distance

$$d(c_i^\alpha, x_i^\alpha, H_i^\alpha) = (x_i^\alpha - c_i^\alpha)^T H_i^{\alpha-1} (x_i^\alpha - c_i^\alpha). \tag{18}$$

Anisotropic mean shift is intended to modulate the kernels during the mean shift procedure. The objective is to keep reducing the Mahalanobis distance so as to group similar samples as much as possible. First, the anisotropic bandwidth matrix H_i^α is estimated with the following constraints:

$$\begin{cases} k_e^\alpha (d(x, x_i, H_i^\alpha)) < 1 \\ k_e^\beta \left(||(x-x_i)/h^\beta (H_i^\alpha)||^2 \right) < 1 \end{cases} \tag{19}$$

The bandwidth matrix can be decomposed to

$$H_i^\alpha = \lambda V A V^T \tag{20}$$

where λ is a scalar, V is a matrix of normalised eigenvectors, and A is a diagonal matrix of eigenvalues whose diagonal elements a_i sum up to 1.

The bandwidth matrix is updated by adding more and more points to the computational list: if these points are similar in colour, then the Mahalanobis distance will be consistently reduced. Otherwise, if the Mahalanobis distance is increased, these points will not be considered in the computation.

Fuzzy c-means clustering is hence combined with an anisotropic mean shift algorithm, and continuously inherits and updates the states, based on the mutual correction of FCM and mean shift.

Table 1 Quantisation results, given in terms of PSNR [dB]

	Lenna	Peppers	Mandrill	Sailboat	Pool	Airplane	average
Popularity alg.	22.24	18.56	18.00	8.73	19.87	15.91	17.22
Median cut	23.79	24.10	21.52	22.01	24.57	24.32	23.39
Octree	27.45	25.80	24.21	26.04	29.39	28.77	26.94
Neuquant	27.82	26.04	24.59	26.81	27.08	28.24	26.73
FCM	28.81	26.77	25.03	27.25	31.03	30.23	28.17
RSFCM	28.70	26.70	24.98	27.32	30.81	30.73	28.20
EnFCM	28.61	26.74	24.87	27.22	31.11	29.92	28.08
AMSFCM	28.63	26.71	24.66	27.24	30.87	29.96	28.01

Anisotropic mean shift based FCM (AMSFCM) proceeds in the following steps:

1. Initialise the cluster centres c_i. Let $j = 0$.
2. Initialise the fuzzy partitions μ_{ij} using Equation (2).
3. Set $j = j + 1$ and compute c_i using Equation (3) for all clusters.
4. Update μ_{ij} using Equation (2).
5. For each pixel x_i determine anisotropic kernel and related colour radius using Equations (17) and (20). Note that mean shift is applied to the outcome image of FCM.
6. Calculate the mean shift vector and then iterate until the mean shift, $M^+(x_i) - M^-(x_i)$, is less than a pixel considering the previous position and a normalised position change:
$$M^+(x_i) = \nu M^-(x_i) + (1-\nu) \cdot \frac{\sum_{j=1}^{N} (x_j - M^-(x_i)) \| (M^-(x_i^\beta) - x_j^\beta)/(h^\beta H_j^\alpha) \|^2}{\sum_{j=1}^{N} \| (M^-(x_i^\beta) - x_j^\beta)/(h^\beta H_j^\alpha) \|^2}$$
with $\nu = 0.5$.
7. Merge pixels with similar colour.
8. Repeat Steps 3 to 7 until convergence.

6 Experimental Results

For our experiments we used six standard images commonly used in the colour quantisation literature (*Lenna, Peppers, Mandrill, Sailboat, Airplane, and Pool*) and applied the different FCM algorithms to generate quantised images with a palette of 16 colours.

To put the results obtained into context, we have also implemented four popular colour quantisation algorithms to generate corresponding quantised images with palette size 16, namely Popularity algorithm [10], Median cut quantisation [10], Octree quantisation [9], and Neuquant [8]. For all algorithms, pixels in the quantised images were assigned to their nearest neighbours in the colour palette to provide the best possible image quality.

The results are listed in Table 1, expressed in terms of peak signal to noise ration (PSNR) defined as

$$\mathrm{PSNR}(I_1, I_2) = 10 \log_{10} \frac{255^2}{MSE(I_1, I_2)} \tag{21}$$

with MSE (the mean-squared error) is calculated as

$$\mathrm{MSE}(I_1, I_2) = \frac{1}{3nm} \sum_{i=1}^{n} \sum_{j=1}^{m} [(R_1(i,j) - R_2(i,j))^2 + (G_1(i,j) - G_2(i,j))^2 + (B_1(i,j) - B_2(i,j))^2] \tag{22}$$

where $R(i,j)$, $G(i,j)$, and $B(i,j)$ are the red, green, and blue pixel values at location (i,j) and n and m are the dimensions of the images.

From Table 1 we can see that of the dedicated colour quantisation algorithms Octree and Neuquant clearly outperform the Popularity and Median Cut methods. For all FCM approaches we ran each algorithm 10 times (randomly initialising the cluster centres) on each image and report the average PSNR of these 10 runs in Table 1. Looking at the results, it is obvious that they all achieve significantly better image quality than any of the other algorithms, including Octree and Neuquant. In fact, on average, FCM colour quantisation provides an increase in PSNR of about 1.5 which is quite remarkable. Also, we can see that the various variants of FCM which provide a signifant speedup in terms of computational requirements follow closely the results of the conventional fuzzy c-means algorithm. This confirms that it is possible to use a more efficient method without decreasing the resulting image quality.

The results shown in Table 1 are further illustrated in Figure 1 which shows a zoomed-in part of the *Pool* image together with the same part extracted from the images colour quantised by all algorithms and the corresponding error images. Error images (or image distortion maps) are commonly employed for judging the difference between images or the performance of competing algorithms [16]. For each quantised images we therefore also provide an error image that represents the difference between the original and the palettised image (the squared error at each pixel location is calculated, the resulting image then inverted and a gamma function applied to increase the contrast). It is clear that the popularity algorithm performs poorly on this image and assigns virtually all of the colours in the palette to green and achromatic colours. Median cut is better but still provides fairly poor colour reproduction; most of the colours in the quantised image are fairly different from the original. The same holds true for the images produced by Neuquant. Here the most obvious artefact is the absence of an appropriate red colour in the colour palette. A far better result is achieved by the Octree algorithm, although here also the red is not very accurate and the colour of the cue is greenish instead of brown. Significantly better image quality is maintained by applying fuzzy c-means clustering. Although the colour palettes have only 16 entries all colours of the original image are accurately presented including the red ball, the colour of the billiard cue as well as good colour reproduction of the green cloth and the reflections on the kettle. This is true not only for the conventional FCM approach but also for the other three, more efficient, derivates.

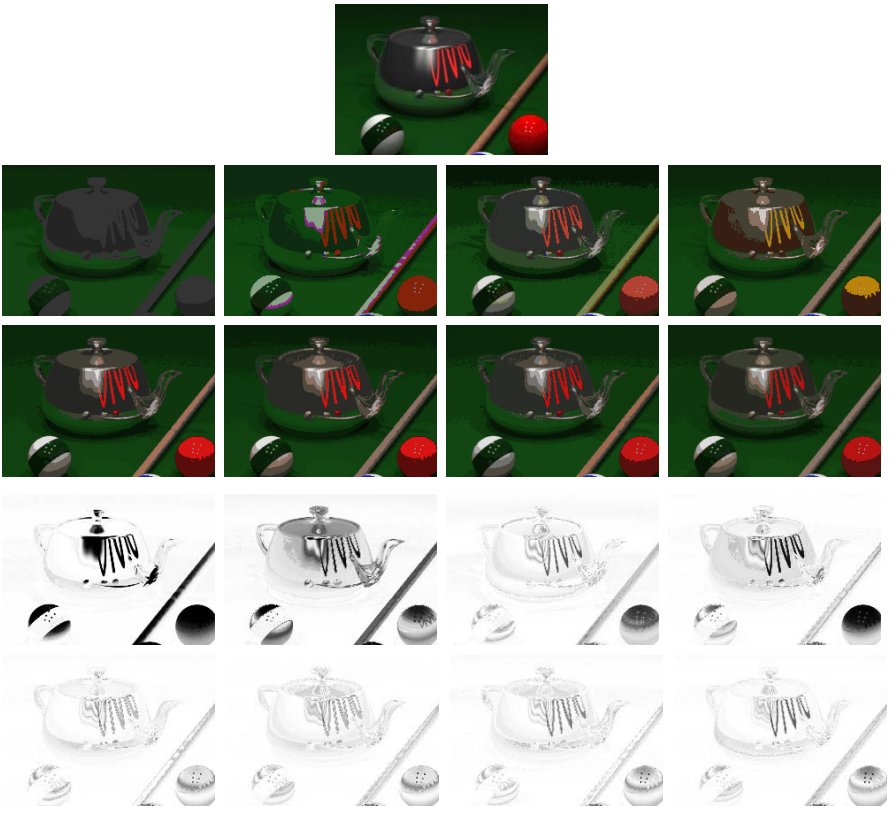


Fig. 1 Part of original *Pool* image (top row) and corresponding images quantised with (from left to right, top to bottom): Popularity, Median cut, Octree, Neuquant; FCM, RSFCM, EnFCM, and AMSFCM algorithms. Also shown are the error images of the quantised images compared to the original

7 Conclusions

In this work we have investigated the performance of fuzzy c-means clustering approaches for colour quantisation. In particular we have evaluated conventional fuzzy c-means, fast fuzzy c-means with random sampling, fast generalised fuzzy c-means, and anisotropic mean shift based fuzzy c-means together with purpose built colour quantisation techniques, namely Popularity algorithm, Median Cut, Octree, and Neuquant. Experimental results have revealed that fuzzy clustering based colour quantisation is able to achieve significantly improved image quality compared to the other techniques. Furthermore, it was shown that the computationally faster variants of fuzzy c-means provide virtually the same image quality as their full conventional counterpart and therefore represent efficient and effective techniques for colour quantisation.

References

1. Ahmed, M., Yamany, S., Mohamed, N., Farag, A., Moriaty, T.: A modified fuzzy c-means algorithm for bias field estimation and segmentation of mri data. IEEE Trans. Medical Imaging 21, 193–199 (2002)
2. Bezdek, J.: A convergence theorem for the fuzzy isodata clustering algorithms. IEEE Trans. Pattern Analysis and Machine Intelligence 2, 1–8 (1980)
3. Cai, W., Chen, S., Zhang, D.: Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. Pattern Recognition 40(3), 825–838 (2007)
4. Cheng, T., Goldgof, D., Hall, L.: Fast fuzzy clustering. Fuzzy Sets and Systems 93, 49–56 (1998)
5. Chuang, K., Tzeng, S., Chen, H., Wu, J., Chen, T.: Fuzzy c-means clustering with spatial information for image segmentation. Computerized Medical Imaging and Graphics 30, 9–15 (2006)
6. Comaniciu, D., Meer, P.: Mean shift analysis and applications. In: 7th Int. Conference on Computer Vision, pp. 1197–1203 (1999)
7. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 603–619 (2002)
8. Dekker, A.H.: Kohonen neural networks for optimal colour quantization. Network: Computation in Neural Systems 5, 351–367 (1994)
9. Gervautz, M., Purgathofer, W.: A simple method for color quantization: Octree quantization. In: Glassner, A.S. (ed.) Graphics Gems, pp. 287–293 (1990)
10. Heckbert, P.S.: Color image quantization for frame buffer display. ACM Computer Graphics (ACM SIGGRAPH 1982 Proceedings) 16(3), 297–307 (1982)
11. Hu, R., Hathaway, L.: On efficiency of optimization in fuzzy c-means. Neural, Parallel and Scientific Computation 10, 141–156 (2002)
12. Nolle, L., Schaefer, G.: Color map design through optimization. Engineering Optimization 39(3), 327–343 (2007)
13. Scheunders, P.: A genetic c-means clustering algorithm applied to color image quantization. Pattern Recognition 30(6), 859–866 (1997)
14. Szilagyi, L., Benyo, Z., Szilagyii, S.M., Adam, H.S.: MR brain image segmentation using an enhanced fuzzy c-means algorithm. In: 25th IEEE Int. Conference on Engineering in Medicine and Biology, vol. 1, pp. 724–726 (2003)
15. Wang, J., Thiesson, B., Xu, Y., Cohen, M.: Image and video segmentation by anisotropic kernel mean shift. In: Pajdla, T., Matas, J.G. (eds.) ECCV 2004. LNCS, vol. 3022, pp. 238–249. Springer, Heidelberg (2004)
16. Zhang, X.M., Wandell, B.A.: Color image fidelity metrics evaluated using image distortion maps. Signal Processing 70(3), 201–214 (1998)
17. Zhou, H., Schaefer, G., Shi, C.: A mean shift based fuzzy c-means algorithm for image segmentation. In: 30th IEEE Int. Conference Engineering in Medicine and Biology, pp. 3091–3094 (2008)

Novel Face Recognition Approach Using Bit-Level Information and Dummy Blank Images in Feedforward Neural Network

David Boon Liang Bong, Kung Chuang Ting, and Yin Chai Wang

Abstract. Bit-level information is useful in image coding especially in image compression. A digital image is constructed by multilevel information of bits, called as bit-plane information. For an 8-bits gray level digital image, bit-plane extraction has ability to extract 8 layers of bit-plane information. Conventional neural network-based face recognition usually used gray images as training and testing data. This paper presents a novel method of using bit-level images as input to feedforward neural network. CMU AMP Face Expression Database is used in the experiments. Experiment result showed improvement in recognition rate, false acceptance rate (FAR), false rejection rate (FRR) and half total error rate (HTER) for the proposed method. Additional improvement is proposed by introducing dummy blank images which consist of plain 0 and 1 images in the neural network training set. Experiment result showed that the final proposed method of introducing dummy blank images improve FAR by 3.5%.

1 Introduction

Conventionally, face recognition is using complex mathematical solution, like eigenfaces [1], LDA [2], Gabor feature [2][3], elastic brunch matching [4], 3-D [5][6][7], wavelet [8][9], and line edge map [10] but all these features involving high complexity of mathematical calculation and they only provides single feature extraction.

Bit-level information is useful in image coding especially in image compression [11][12]. Digital image compression method is based on concept of decomposing a multilevel image into a series of binary images and compressing each binary image via one of several well-known binary compression methods [13]. A digital image is constructed by multilevel information of bits, called as bit-plane information.

Since bit-planes are information extracted from binary code of image pixels, they contain useful raw data of image. In this paper, we proposed a novel method of using bit-level images to act as feature extraction input to feedforward neural

David Boon Liang Bong · Kung Chuang Ting
Faculty of Engineering, Universiti Malaysia Sarawak, 94300 Kota Samarahan, Malaysia

Yin Chai Wang
Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, 94300 Kota Samarahan, Malaysia

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 483–490.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

network. This novel method have advantages in feature extraction compare to conventional methods because it extracted multiple features data in single image, and reduced the complexity of computer calculation.

2 Bit Level Information

A digital image consists of m x n pixels and represented by gray-level value for each pixel. Normally, the gray level range is from 0 to 255. Digital computers only manipulate true and false signals and therefore the gray-level range is represented by 8 bits data, which form bit-plane 0 until bit-plane 7. Bit-plane 0 represents least significant bit while bit- plane 7 is the most significant bit. In term of 8-bit bytes, bit-plane 0 contains all the lowest order bits in the pixels of the image and bit- plane 7 contains all the highest order bits [13]. The higher-order bits, bit-plane 4 until bit- plane 7, contain majority of visually significant data, and other lower-order bits, bit-plane 0 until bit-plane 3, contain more subtle details in the image [13].

Fig. 1 shows an example of the extraction process of 7 bit-planes information from a digital image.

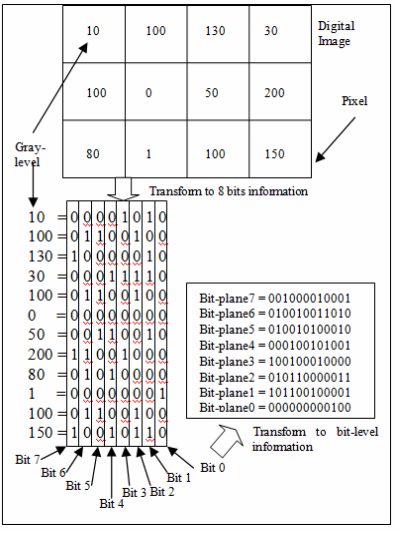


Fig. 1 Example of bit-plane extraction process from a digital image

Bit-planes can be constructed from a gray-level image as follows:

i) Original image,

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix} \quad (1)$$

where *f(x,y)*= original image

ii) Bit-plane 0

$$f_{bp\,0}(x,y) = \begin{bmatrix} R(\frac{f(0,0)}{2}) & R(\frac{f(0,1)}{2}) & \cdots & R(\frac{f(0,N-1)}{2}) \\ R(\frac{f(1,0)}{2}) & R(\frac{f(1,1)}{2}) & \cdots & R(\frac{f(1,N-1)}{2}) \\ \cdots & \cdots & \cdots & \cdots \\ R(\frac{f(M-1,0)}{2}) & R(\frac{f(M-1,1)}{2}) & \cdots & R(\frac{f(M-1,N-1)}{2}) \end{bmatrix} = R(\frac{f(x,y)}{2}) \quad (2)$$

iii) Bit-plane 1

$$f_{bp1}(x,y) = \begin{bmatrix} R(\frac{floor(\frac{f(0,0)}{2})}{2}) & R(\frac{floor(\frac{f(0,1)}{2})}{2}) & \cdots & R(\frac{floor(\frac{f(0,N-1)}{2})}{2}) \\ R(\frac{floor(\frac{f(1,0)}{2})}{2}) & R(\frac{floor(\frac{f(1,1)}{2})}{2}) & \cdots & R(\frac{floor(\frac{f(1,N-1)}{2})}{2}) \\ \vdots & \vdots & \cdots & \vdots \\ R(\frac{floor(\frac{f(M-1,0)}{2})}{2}) & R(\frac{floor(\frac{f(M-1,1)}{2})}{2}) & \cdots & R(\frac{floor(\frac{f(M-1,N-1)}{2})}{2}) \end{bmatrix} = R\left[\frac{1}{2} floor\left(\frac{1}{2}[f(x,y)]\right)\right] \quad (3)$$

iv) For bit-plane 2 until bit-plane 7, calculation is repeated as (iii), and general formula is formed for bit-plane construction as:

$$f_{bp_i}(x,y) = R\left[\frac{1}{2} floor\left(\frac{1}{2^i}[f(x,y)]\right)\right], \text{ i=0,1,…,7} \quad (4)$$

where f(x,y) = original image, $f_{bp}(x,y)$ = bit-plane information, R = remainder, and floor(x) = round the elements to x nearest integers less than or equal to x.

3 Feedforward Neural Network

A multi-layer feedforward neural network will be used to model the recognition system. Feedforward neural network [14] has two stage, learning stage and testing stage. Major function of learning stage is updating the weight of networks, and for testing stage is evaluating the accuracy and reliability of the networks.

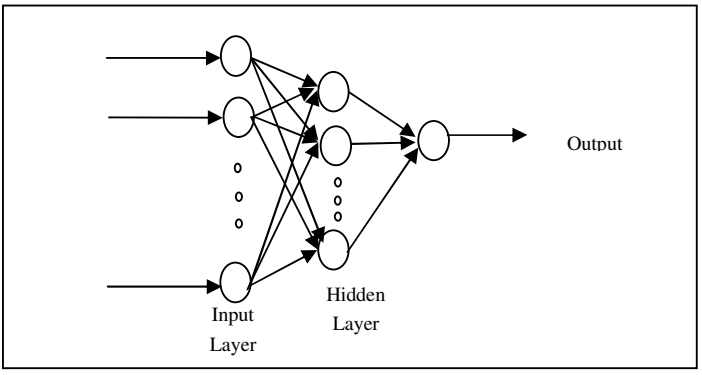


Fig. 2 Architecture of feedforward neural networks

4 Face Expression Database

The database used for evaluation is CMU AMP Face Expression Database, which consists of 13 subjects, and each with 75 images [15]. All these images are in gray values and have resolution of 4096 (64 x 64) pixels. Some samples of this face database are shown in Fig. 3.

Fig. 3 Samples of CMU AMP face expression database



5 Dummy Blank Images

Dummy blank images are images which do not carry any information. Two different dummy blank images used are dummy blank white image with all the pixels set to 1, and dummy blank black image with all the pixels set to 0. These two dummy blank images are added into feedforward neural networks learning stage. Fig. 4 shows the learning stages of feedforward neural networks without and with blank images respectively.

The system is trained using three-layer feedforward neural networks algorithm. These three layers are the input layer, hidden layer and output layer. Input layer consists of 4096 neurons which is the dimension of image's pixels. Input data is bit-level information of the image as described in section 2. The second layer is hidden layer which consists of the same number of neurons with input layer - 4096 hidden neurons. These hidden neurons are connected with 4 output neurons in the output layer.

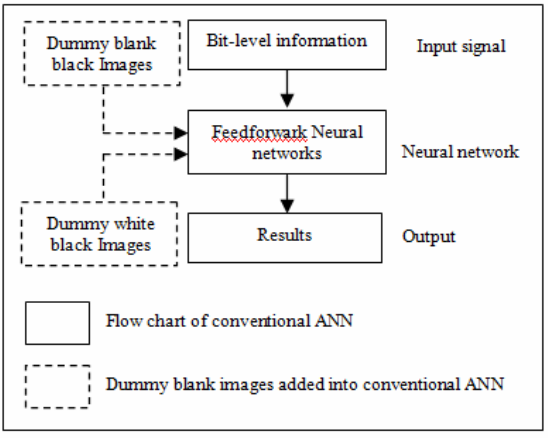


Fig. 4 Dummy blank images added into feedforward neural networks' learning stage

Feedforward neural network is a type of supervised neural networks. It needs target outputs during the learning stage. Target outputs are trained from 0001, 0010, … , 1101 respectively according to face database ID (ID 1 until 13). For added dummy blank images added, 0000 (or ID 0) is the target output, while 1111 (ID 15) is the target output of dummy blank white image. Additional of dummy blank images not only modified the weights of hidden neurons, but also creates two more classes for neural networks' classification. That is the reason for the improved ability to identify impostors.

6 Data and Analysis

Performances of system had been tested and analyzed. Some parameters are set during experiments to ensure that comparison is done under a constant situation. Training samples used in feedforwak neural network are 14 images per subject and same training samples are used throughout all experiments. Table 1 shows the performance of recognition rates of combined bit-planes and gray-level in ANN which trained with and without dummy images. Combined bit-planes consist of bit-plane 4, 5, 6 and 7, and these 4 bit-planes are trained and have individual output. All these individual outputs will compare against one another to generate a final output. Final output only recognizes a client if majority (more than 50%) of bit-planes gives same result, otherwise input is treated as impostor. From Table 1, recognition rates of using combined bit-planes are always better than gray-level regardless of the ANN trained with or without dummy images. Performance of the combined bit-planes is increased by 0.4% after ANN is trained with dummy images compare to ANN trained without dummy images.

People who are not client in the system database, but trying to access into the system without authority are called as an impostor. A reliable recognition system should be able to reject impostors successfully. If a system is unable to reject impostors, unauthorized access will be granted and this posed a serious weakness of the system. Determination of the efficiency of a recognition system to reject impostors is by using false acceptance rate (FAR).

Table 1 Recognition rates for combined bit-planes and gray-level

Bit-plane	Recognition Rates without impostors (ANN trained without dummy images)	Recognition Rates without impostors (ANN trained with dummy images)
Combined bit-planes	96.2%	96.6%
Gray-level	82.3%	77.9%

Table 2 FAR, FRR and HTER for ANN trained with and without dummy blank inages

Bit-plane	**ANN trained without dummy blank images**			**ANN trained with dummy blank images**		
	FAR %	**FRR %**	**HTER %**	**FAR %**	**FRR%**	**HTER%**
combined bit-planes	26.2	2.2	14.2	22.7	2.5	12.6
gray-level	81.3	11.5	46.4	70.6	7.6	39.1

$$FAR = \frac{number\ \ of\ \ FAs}{number\ \ of\ \ impostor\ \ accesses} \times 100\% \tag{5}$$

Besides FAR, another rate is also calculated, which is false rejection rate (FRR). FRR is used as determination of the efficiency of a recognition system to accept clients of the system.

$$FRR = \frac{number\ \ of\ \ FRs}{number\ \ of\ \ client\ \ access} \times 100\ \% \tag{6}$$

In order to gauge the performance of the system with regards to both FAR and FRR, half total error rate (HTER) is used. HTER is basically the average of both FAR and FRR as in equation (7).

$$HTER = \frac{FAR + FRR}{2} \tag{7}$$

From table 2, after dummy blank images are added into learning stage of the system, it enhanced the recognition rate of the system regardless of using either gray-level or combined bit-level images. The addition of dummy blank images reduced FAR by 3.5% for combined bit-planes and 10.7% for gray-level images.

HTER indicated the error rate of using combined bit-levels is much lower than gray-level, with or without dummy blank images. After dummy blank images are added into neural networks training, HTER of combined bit-planes improved by 1.6%, and for gray-level, it also improved by 7.3%.

From these results, dummy blank images play an important role in neural network's learning stage in order to increase the performance of the system. By adding dummy blank images, ANN had modified the weights of hidden neurons to differentiate between clients and impostors.

7 Conclusion

Bit-level information is reliable information for pattern recognition. After these four bit-planes are combined, its performance is better than using gray-level images. Gray-level images give higher FAR and FRR compare to combined bit-levels. Higher FAR and FRR in a system is a threat to security because it could not reject impostors efficiently.

Dummy blank images are images which do not carry any information. Dummy blank images added during ANN learning stage will modify the weights of hidden neurons and its results show better manipulation of ANN for output. Experiment results showed that the final proposed method of introducing dummy blank images improve HTER by 1.6% for combined bit-planes.

Acknowledgments. The authors would like to express their appreciation to Universiti Malaysia Sarawak for the support given to this project, and Advanced Multimedia Processing (AMP) Lab, Carnegie Mellon University for sharing the face expression database.

References

1. Turk, M.A., Pentland, A.P.: Face Recognition Using Eigenfaces. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Hawaii, pp. 586–591 (1991)
2. Cruz-Llanas, S., Ortega-Garcia, J., Martinez-Torrico, E., Gonzalez-Rodriguez, J.: Comparison of Feature Extraction Techniques in Automatic Face Recognition Systems for Security Application. In: IEEE 34th Annual International Carnahan Conference on Security Technology, Ottawa, pp. 40–46 (2000)
3. Feris, R.S., Cesar, R.M., Kruger, V.: Efficient Real-Time Face Tracking in Wavelet Subspace. In: IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, Vancouver, pp. 113–118 (2001)
4. Wiskott, L., Fellous, J.M., Kuiger, N., von der Malsburg, C.: Face Recognition by Elastic Bunch Graph Matching. IEEE Trans. Pattern Analysis and Machine Intelligence 19(7), 775–779 (1997)
5. Lao, S., Sumi, Y., Kawade, M., Tomita, F.: 3D Template Matching for Pose Invariant Face Recognition Using 3D Facial Model Built with Isoluminance Line Based Stereo Vision. In: 15th International Conference on Pattern Recognition, Barcelona, vol. 2, pp. 911–916 (2000)
6. Hu, Y., Jiang, D., Yan, S., Zhang, L., Zhang, H.: Automatic 3D Reconstruction for Face Recognition. In: Sixth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 843–848 (2004)
7. Bowyer, K.W., Chang, K., Flynn, P.: A Survey of Approaches to Three-Dimensional Face Recognition. In: 17th International Conference on Pattern Recognition, vol. 1, pp. 358–361 (2004)
8. Zhang, B., Zhang, H., Sam Ge, S.: Face Recognition by Applying Wavelet Subband Representation and Kernel Associative Memory. IEEE Transactions on Neural Networks 15(1), 166–177 (2004)

9. Amira, A., Farrell, P.: An Automatic Face Recognition System Based on Wavelet Transforms. In: IEEE International Symposium on Circuits and Systems 2005, vol. 6, pp. 6252–6255 (2005)
10. Gao, Y.S., Leung, M.K.H.: Face Recognition Using Line Edge Map. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(6), 764–779 (2002)
11. Rabbani, M., Melnychuck, P.W.: Conditioning Contexts for The Arithmetic Coding of Bit Planes. IEEE Trans. on Signal Processing 40(1), 232–236 (1992)
12. Zhang, R., Yu, R., Sun, Q., Wong, W.: A New Bit-Plane Entropy Coder for Scalable Image Coding. In: IEEE International Conference on Multimedia and Expo 2005, pp. 237–240 (2005)
13. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice Hall, Inc., New Jersey (2002)
14. Fausett, L.: Fundamentals of Neural Networks: Architectures, Algorithms, and Applications. Prentice Hall, New Jersey (1994)
15. AMP Advance Multimedia Processing Lab, Face Authentication Project, `http://amp.ece.cmu.edu/projects/FaceAuthentication/download.htm`

ICA for Face Recognition Using Different Source Distribution Models

Dinesh Kumar, C.S. Rai, and Shakti Kumar

Abstract. Independent Component Analysis (ICA) is a method for separating the components as independent as possible, out of mixture of a number of mixed signals. There exist a number of statistical techniques based on information theoretic concepts and algebraic approaches for performing ICA. Neural algorithms derived from these approaches are used for extracting the independent components. Besides its application in signal processing such as speech signal separation, it has also successfully been used for face recognition. Analysis of face database has shown that the face images do possess negative and positive kurtosis values. This indicates that face images are of sub-Gaussian and super-Gaussian nature. This paper proposes a technique that deals with such type of face database. In this work, based on the assumption of the nonlinearity for sub-Gaussian sources, the nonlinearity for the super-Gaussian sources has been derived. A symmetric probability density function around the Gaussian distribution has been used for representing the sub and super-Gaussian sources. The proposed method has been successfully used for face recognition. Experimental results with the ORL face database have been presented. The results show the superiority of the proposed algorithm.

Dinesh kumar
Department of Computer Science and Engineering,
Guru Jambheshwar University of Science and Technology,
Hisar - 125001, Haryana, India
e-mail: dinesh_chutani@yahoo.com

C.S. Rai
University School of Information Technology,
GGS Indraprastha University, Delhi, India
e-mail: csrai_ipu@yahoo.com

Shakti kumar
Institute of Science and Technology Klawad,
Distt. Yamuna Nagar, Haryana, India
e-mail: shakti@istk.org

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 491–498.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

1 Introduction

Principal Component Analysis (PCA), a successful and popular unsupervised technique has widely been used for useful image representations. This method deals with the second order statistics and it gives a new set of basis images. The image coordinates (PCA coefficients) are uncorrelated in this new basis. PCA exploits only the pairwise relationship between the pixels in the face image database. Independent Component Analysis (ICA) is a generalization of PCA in which higher order statistics are also considered. It is expected to give better basis images due to the fact that this method is sensitive to higher ordered dependencies. ICA was initially used for what is known as Blind Source Separation (BSS)[1][3][4][6][8]. A number of algorithms for performing ICA have been reported in the literature [9]. In this paper the following generative model of ICA has been used.

Consider a random vector $x = (x_1, x_2, \ldots, x_n)^T$ and the components of the random vector as $s = (s_1, s_2, \ldots, s_n)^T$. The aim is to find the components s_i as independent as possible in the sense of maximizing some function $F = (s_1, s_2, \ldots, s_n)$ that measures independence. The independent component analysis of the observed data x consists of finding a linear transformation

$$s = Wx \tag{1}$$

so as to get the components s_i as independent as possible. This is the most general definition of independent component analysis. We consider simplified model

$$x = As \tag{2}$$

where s is as defined above and the matrix A is a constant $n \times n$ mixing matrix.To recover the individual components, we need $n \times n$ matrix W such that $y = Wx$ where $y = (y_1, y_2, \ldots, y_n)^T$ is an n-dimensional output vector. Components of y must be as close to s as possible. There exist methods for separating mixture of sub-Gaussian and super-Gaussian signals. These methods have been applied for separating the independent components from a mixture of audio signals [10][12]. ICA has also been applied for applications like image separation [5], face recognition [2] etc. Two architectures [2] were proposed for representing the face images. The assumption is that the face images in x are a linear mixture of an unknown set of statistically independent source images s and A represents unknown mixing matrix. The sources are recovered by a matrix of learned filters W_I, which produce statistically independent outputs U. In architecture I, used in this paper for representing face images, the rows of the input matrix x contained the images. The ICA outputs in the rows of $W_I x = U$ were also images and provided a set of independent basis images for the faces. The ICA representation consisted of the coefficients for the linear combination of independent basis images in U that comprised each face image. The number of independent components was controlled by performing ICA on a set of m linear combinations of those images where $m < n$ (number of original images). A matrix (ϕ_m) consisting the first m principal component vectors of the image set was

considered and ICA was performed on ϕ_m^T resulting in a matrix of m independent source images in the rows of U. The calculations for the coefficients are as follows:

The PC representation is defined as $R_m = x \times \phi_m$ and $x_{rec} = R_m \times \phi_m^T$ is the minimum squared error approximation of x. The ICA algorithm produced a matrix $W_I = W \times W_z$ such that $W_I \times \phi_m^T = U \Rightarrow \phi_m^T = W_I^{-1} \times U$, $W_z = 2\times < xx^T >^{-1/2}$. Therefore $x_{rec} = R_m \times \phi_m^T \Rightarrow x_{rec} = R_m \times W_I^{-1} U$. Hence the rows of $B = R_m \times W_I^{-1}$ contained the coefficients for the linear combination of statistically independent sources U and comprised x_{rec} where x_{rec} was the minimum squared error approximation of x, just as in PCA. In order to obtain the representation for test images we first obtain $R_{test} = x_{test} \times \phi_m$ and then compute $B_{test} = R_{test} \times W_I^{-1}$.

2 Algorithm

The objective function, $\psi(W)$ for the source separation can be expressed as [7]

$$\psi(W) = -log|det(W)| - \sum_{i=1}^{n} log f_i(y_i) \tag{3}$$

where $f_i(y_i)$ are the marginal pdfs of the output components y_i. The minimization of the objective function as given by (3) leads to the following [1][7]

$$W(k+1) = W(k) + \eta(k)[I - \phi(y(k))y^T(k)]W(k) \tag{4}$$

where k denotes the time index, η is the learning rate parameter, and $\phi(y) = [\phi(y_1), \phi(y_2), \ldots, \phi(y_n)]^T$ is the component wise nonlinear function. The ith nonlinear function $\phi(y_i)$ is expressed as [7]

$$\phi(y_i) = \frac{\partial log f_i(y_i)}{\partial y_i} \tag{5}$$

3 Probability Density Functions

The mixture of signals contains signals either of sub-Gaussian or super-Gaussian type. The sub-Gaussian distributions have flatter tail than the Gaussian type distributions. Whereas the other type of distributions (super-Gaussian) has relatively large tails and sharp peaks as compared to Gaussian distribution. Kurtosis is a measure of nongaussianity of a random variable. It is defined as

$$kurt(y_i) = \frac{E(y_i^4)}{(E(y_i^2))^2} - 3 \tag{6}$$

where $kurt(y_i)$ is the kurtosis of the ith component y_i. Super-Gaussian signals have positive kurtosis and are also known as leptokurtic in statistical literature

whereas sub-Gaussian signals have negative value of kurtosis and are also known as platykurtic[9].

Let us consider that the probability density function (pdf) of a Gaussian distributed random variable y is denoted as $f(y)$. Let us further denote $f_{super}(y)$ and $f_{sub}(y)$ as the pdfs of super and sub Gaussian distributions respectively. The main objective here is to presume a suitable choice of nonlinearity for the sub-Gaussian type distribution and then to find the probability distributions symmetrical around the Gaussian distribution and hence to derive the nonlinearity for super-Gaussian distribution. The symmetry around the Gaussian distribution means

$$2f(y) = f_{super}(y) + f_{sub}(y) \tag{7}$$

i.e., the average of pdfs of super and sub-Gaussian distributions equals Gaussian distribution for all values of y and

$$f_{sub}(y) = 2f(y) - f_{super}(y) \tag{8}$$

Hence for a known value of one pdf, we can easily find another. We assume that following density model has been used for super-Gaussian sources

$$f_{super}(y) \propto f(y_i)g(y_i) \tag{9}$$

where $f(y_i)$ is a Gaussian density function with zero mean and unit variance and is equal to $(1/\sqrt{2\pi})e^{-y_i^2/2}$ and $g(y_i)$ is a function that is to be determined. From (8) and (9), $f_{sub}(y)$ can be expressed as

$$f_{sub}(y) = 2f(y) - f(y)g(y) \tag{10}$$

For computational simplicity, the constant of proportionality in (9) is considered as unity. We further presume that y_i^3 has been used as the nonlinearity for the sub-Gaussian distributions as it has been found in literature that this is also one of the types of nonlinearities commonly used for sub-Gaussian distributions. Using (5), we can write

$$\phi(y_i) = \frac{\partial log f_{sub}(y_i)}{\partial y_i} = y_i^3 \tag{11}$$

Equation (11) can now be written as

$$\frac{\partial log[2f(y_i) - f(y_i)g(y_i)]}{\partial y_i} = y_i^3 \tag{12}$$

Simplification gives

$$g(y_i) = 2 - (\sqrt{2\pi})e^{y_i^2/2}e^{y_i^4/4} \tag{13}$$

So

$$f_{super}(y_i) = f(y)(2 - (\sqrt{2\pi})e^{y_i^2/2}e^{y_i^4/4}) \tag{14}$$

The nonlinearity corresponding to pdf given by (14) can be written as

$$\phi(y_i) = \frac{\partial log f_{super}(y_i)}{\partial y_i} = \frac{\partial}{\partial y_i}(log(f(y_i)[2 - ((\sqrt{2\pi})e^{y_i^2/2}e^{y_i^4/4})])) \quad (15)$$

this simplifies to

$$\phi(y_i) = -4y - 5y^3e^{y^2/2}e^{y^4/4} + 5ye^{y^2/2}e^{y^4/4} + 6.25y^3e^{y^2}e^{y^4/2} \quad (16)$$

When the value of the kurtosis is positive, the nonlinearity given by (16) is used and if the kurtosis is negative, (11) is used as the nonlinearity.

4 Experimental Results

For our experimental work, we used ORL face database. It is composed of 400 images with each image having a resolution 92×112. As many as 40 different people (subjects/classes) are contained in the database and each subject has his/her 10 different images. Figure 1 shows some sample images of this database. These images vary in terms of facial expressions and facial details. For computational simplicity, the original image 92×112 was resized to 60×60 prior to further processing of the face image.



Fig. 1 Some images of ORL face database

For the experimental work, six sets (Table 1) of training and testing images were used. Each set has 5 training images and 5 testing images. The training and testing images are nonoverlapping.

The matching of test images was done using Euclidean norm (L2 norm) as the distance measure. ICA was performed on the matrix x containing the first forty percent of the Principal Component axes of total number of training images arranged in rows accounting for a total variance of almost 90 percent (Table 2). The table shows the values for 100 images (20 classes). The Total Variance Contribution Rate (TVCR) was calculated as

$$TVCR(\lambda_1, \lambda_2, \ldots, \lambda_d) = \frac{\sum_{i=1}^{d} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \quad (17)$$

Table 1 Training and Testing sets

Set Number	Training Images	Test Images
1	1,2,3,4,5	6,7,8,9,10
2	2,3,4,5,6	1,7,8,9,10
3	3,4,5,6,7	1,2,8,9,10
4	4,5,6,7,8	1,2,3,9,10
5	5,6,7,8,9	1,2,3,4,10
6	6,7,8,9,10	1,2,3,4,5

where λ_i is eigenvalue corresponding to the ith eigenvector obtained by applying Principal Component Analysis (PCA) on the input face image data matrix. The d represents the number of eigenvalues selected out of total of p eigenvalues.

Table 2 Total Variance Contribution Rate (TVCR) for Different Number of Eigenvalues

Number of Eigenvalues(d)	99	90	80	60	40	20	10
TVCR	100	99.5	98.6	95.6	90.3	78.6	63.8

Prior to performing ICA, the input data was whitened by passing x through the whitening matrix $2 \times (Cov(x))^{-1/2}$ thus removing the first and second order statistics of data. The ICA was performed. The weights W were updated according to (4) for 100 iterations. The learning rate was chosen as 0.000001. The experiment was performed for 10, 20 and 40 classes. Each class has 10 images out of which 5 images from each class were used for training and the remaining 5 as the test set (Table 1). The recognition rate was calculated. Table 3 shows the results for different classes. The results were compared with the Infomax algorithm [3] for ICA. The nonlinearity used was the logistic function given as

$$Z_i = \frac{1}{1+e^{-y_i}}, i = 1,2,\ldots,n \tag{18}$$

for an n-dimensional random vector X representing the distribution of the inputs and W being the $n \times n$ invertible matrix, $Y = WX$ and $Z = f(Y)$, an n-dimensional random variable representing the outputs of n-neurons and the weight update rule is given as [3]

$$\Delta W = \eta(I + (1-2Z)Y^T)W \tag{19}$$

The weights W were updated according to (19) for 1600 iterations. The learning rate was initialized at 0.001 and annealed down to 0.0001. The Euclidean norm (L2 norm) was used as the distance measure.

The average recognition rates for 10, 20 and 40 classes are 92%, 90.33%, 90.25% and 92%, 90.17%, 89% for the proposed method and logistic function using Infomax algorithm respectively. Table 3 shows that for different sets we get different

Table 3 Recognition Rate for Different Number of Classes

Number of Classes	10		20		40	
Nonlinearity→	Proposed Logistic Fn		Proposed Logistic Fn		Proposed Logistic Fn	
Set Number	Recognition Rate					
1	96.00	96.00	91.00	91.00	88.00	85.50
2	94.00	94.00	94.00	94.00	92.50	92.00
3	92.00	92.00	94.00	94.00	93.00	92.50
4	86.00	86.00	86.00	85.00	89.50	88.00
5	90.00	90.00	89.00	89.00	89.50	89.00
6	94.00	94.00	88.00	88.00	89.00	87.00

values of recognition rates. This indicates that the images constituting the training and test sets do affect the performance of the face recognition system in terms of recognition rate. The recognition rate for 10 classes is more and it decreases as the number of classes increase. The reason for this may be that as the number of classes and hence the number of images increases, the chances for mismatch are also increased and this results in decrease in the recognition rate.

5 Conclusions

In this paper, a technique for selecting suitable density functions for face recognition was proposed. Based on the presumption that the nonlinearity for sub-Gaussian signals is known, the probability density functions for both sub and super-Gaussian distributions were found with respect to Gaussian distribution and the nonlinearity for the super-Gaussian signals was derived. Simulations for face recognition show the usefulness of the proposed technique. The results show the superiority of the proposed algorithm over Infomax algorithm using logistic function as the nonlinearity. The results also reveal that the images constituting the training and test sets do affect the performance of the face recognition system in terms of recognition rate.

References

1. Amari, S.I., Cichoki, A., Yang, H.: A New Learning Algorithm for Blind Signal Separation. Adv. Neural Information process. Systems 8, 757–763 (1996)
2. Bartlett, M.S., Movellan, J.R., Sejnowski, T.J.: Face Recognition by Independent Component Analysis. IEEE Trans. on Neural Networks 13(6), 1450–1464 (2002)
3. Bell, A.J., Sejnowski, T.J.: An Information Maximization Approach to Blind Separation and Blind Deconvoution. Neural Computations 7(6), 1129–1159 (1995)
4. Cardoso, J.F., Laheld, B.H.: Equivalent Adaptive Source Separation. IEEE Trans. Signal Processing 44(12), 3017–3030 (1996)

5. Cichocki, A., Kasprzak, W.: Local Adaptive Learning Algorithms for Blind Separation of Natural Images. Neural Network World 6(4), 515–523 (1996)
6. Comon, P.: Independent Component Analysis-A New Concept. Signal Processing 36(3), 287–314 (1994)
7. Douglas, S.C., Amari, S.I.: Natural Gradient Adaptation. In: Haykin, S. (ed.) Unsupervised Adaptive Filtering, vol. 1, pp. 13–61. Wiley, New York (2000)
8. Haverinen, A., Oja, E.: Fast Fixed Point Algorithm for Independent Component Analysis. Neural Computation 9(7), 1483–1492 (1997)
9. Haverinen, A., Oja, E., Karhunen, J.: Independent Component Analysis. Wiley, New York (2001)
10. Lee, T.W., Girolami, M., Sejnowski, T.J.: Independent Component Analysis using an Extended Infomax Algorithm for Mixed Sub-Gaussian and Super-Gaussian Sources. Neural Computation 11(2), 417–441 (1999)
11. ORL Face Database, `http://www.cl.cam.ac.uk/ORLfacedatabase`
12. Rai, C.S., Yogesh, S.: Source Distribution Models for Blind Source Separation. Neurocomputing 57, 501–504 (2004)

Object Recognition Using Particle Swarm Optimization on Moment Descriptors

Muhammad Sarfraz and Ali Taleb Ali Al-Awami

Abstract. This work presents study and experimentation for object recognition when isolated objects are under discussion. The circumstances of similarity transformations, presence of noise, and occlusion have been included as the part of the study. For simplicity, instead of objects, outlines of the objects have been used for the whole process of the recognition. Moment Descriptors have been used as features of the objects. From the analysis and results using Moment Descriptors, the following questions arise: What is the optimum number of descriptors to be used? Are these descriptors of equal importance? To answer these questions, the problem of selecting the best descriptors has been formulated as an optimization problem. Particle Swarm Optimization technique has been mapped and used successfully to have an object recognition system using minimal number of Moment Descriptors. The proposed method assigns, for each of these descriptors, a weighting factor that reflects the relative importance of that descriptor.

1 Introduction

In many image analysis and computer vision applications, object recognition is the ultimate goal. Several techniques [3-13] have been developed that derive invariant features from moments for object recognition and representation. These techniques are distinguished by their moment definition, such as the type of data exploited and the method for deriving invariant values from the image moments. This work presents study and experimentation for object recognition when isolated objects are under discussion. The circumstances of similarity transformations, presence of noise, and occlusion have been included as the part of the study. For simplicity, instead of objects, outlines [17] of the objects after edge detection [14-16] have been used for the whole process of the recognition. Hu's moments and their extended counterparts [1-6] have been used as features of the objects. Various similarity measures have been used and compared for recognition process. The test objects are matched with the model objects in database and the object

Muhammad Sarfraz
Department of Information Science, Adailiyah Campus, Kuwait University,
P.O. Box 5969, Safat 13060, Kuwait
e-mail: prof.m.sarfraz@gmail.com, sarfraz@cfw.kuniv.edu

Ali Taleb Ali Al-Awami
Department of Electrical Engineering, King Fahd University of Petroleum and Minerals,
Dhahran, 31261 Saudi Arabia

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 499–508.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

with the least similarity measure is taken as the recognized object. A detailed experimental study has been made under different conditions and circumstances.

From the analysis and results using Moment Descriptors, the following questions arise: What is the optimum number of descriptors to be used? Are these descriptors of equal importance? To answer these questions, the problem of selecting the best descriptors has been formulated as an optimization problem. The idea of using the Particle Swarm Optimization (PSO) methodology [19-23] has been explored. Particle Swarm Optimization technique was used in [23] for the optimization of Fourier descriptors. This study has been mapped and used successfully to have an object recognition system using minimal number of moment descriptors. The goal of the proposed optimization technique is to select the most helpful moment descriptors that will maximize the recognition rate. The proposed method will assign, for each of these descriptors, a weighting factor that reflects the relative importance of that descriptor.

The outline of the remainder of the paper is as follows. Getting of bitmap images and their outline is discussed in Sections 2 and 3 respectively. Section 4 deals with moments. The concept of similarity measures are explained in Section 5. Particle Swarm Optimization technique has been explained, mapped, used, experimented, and analyzed in Section 6. Finally, Section 7 concludes the paper as well as touches some future work.

2 Getting Bitmap Image

Bitmap image of a character can be obtained by creating a bitmap character on some program like *Paint* or Adobe Photoshop. Alternatively an image drawn on paper can scan and store it as bitmap. We used both methods. The quality of bitmap image obtained directly from electronic device depends on the resolution of device, type of image (e.g. bmp, jpeg, tiff etc), number of bits selected to store the image etc. The quality of scanned image depends on factors such as quality of image on paper, scanner and attributes set during scanning. Figure 1(a) shows the bitmap image of a character.

3 Finding Boundary

In order to find boundary of bitmap image, first its chain code is extracted [14, 15]. Chain codes are a notation for recording the list of edge points along a contour. The chain code specifies the direction of a contour at each edge in the edge. From chain coded curve, boundary of the image is found [16]. The selection of Boundary Points is base on their corner strength and contour fluctuations. The input to our boundary detection algorithm is a bitmap image. The algorithm returns number of pieces in the image and for each piece number of Boundary Points and values of these Boundary Points $P_i = (x_i, y_i),\ i = 1,2,...,N$. Figure 1(b) shows detected boundary of the image of Figure 1(a).

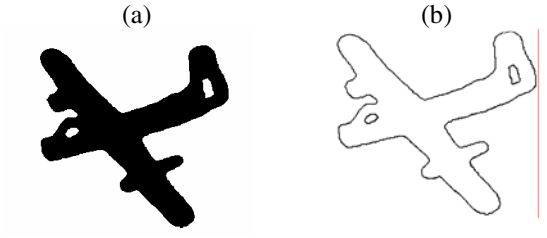


Fig. 1 (a) Bitmap image, (b) Outline of the image

4 Moments

Traditionally, moment invariants are computed based on the information provided by the shape boundary as well as its interior region [1]. The moments used to construct the moment invariants are defined in the continuous form but for practical implementation they are computed in the discrete form. For the detailed description of the moments, the author is referred to [1, 9, 12-13].

5 Similarity Measure

Given two sets of descriptors, how do we measure their degree of similarity? An appropriate classification is necessary if unknown shapes are to be compared to a library of known shapes. If two shapes, A and B, produce a set of values represented by $a(i)$ and $b(i)$ then the distance between them can be given as $c(i) = a(i) - b(i)$. If $a(i)$ and $b(i)$ are identical then $c(i)$ will be zero. If they are different then the magnitudes of the components in $c(i)$ will give a reasonable measure of the difference. It proves more convenient to have one value to represent this rather than the set of values that make up $c(i)$. The easiest way is to treat $c(i)$ as a vector in a multi-dimensional space, in which case its length, which represents the distance between the objects, is given by the square root of the sum of the squares of the elements of $c(i)$.

Various similarity measures may be attempted for experimental studies. For simplicity, we are going to choose Euclidean Distance (ED) measure as a similarity measure for this paper. It provides promising results and is given as follows:

$$\sqrt{\sum_{i=1}^{n} c(i)^2}$$

In this study, n is the number of moments considered, $a(i)$ is the ith moment of the template image, and $b(i)$ is the ith moment of the test image. A tolerable threshold ρ is selected to decide a test object recognized. This threshold is checked against the least value of the selected similarity measure.

The recognition system is tested by generating the test objects by translating, rotating, scaling, adding noise, and adding occlusion to the model objects

contained in a database of different sizes. The test objects were randomly rotated, translated, and scaled. Some were considered without scale of their model sizes. About 100 test objects were used for each of the experiments for testing similarity transformation. The salt & pepper noise [17-18] of different densities is added to the objects for generating the noisy test objects. Median filter was used in the experiment to filter the noise, so that the noise remains on the boundary of the object.

The detailed experimentation has been reported in [9, 23]. One can see that Moment invariants are not promising in general and, for the recognition of occluded objects, the results are not good in particular.

6 Optimization Using Particle Swarm

From the previous analysis and results, the following questions arise: What is the optimum number of descriptors to be used? Are these descriptors of equal importance? To answer these questions, the problem of selecting the best descriptors can be formulated as an optimization problem. The goal of the optimization is to select the most helpful descriptors that will maximize the recognition rate and to assign for each of these descriptors a weighting factor that reflects the relative importance of that descriptor. A novel population based optimization approach, called Particle Swarm Optimization (PSO), has been chosen to goal. The details of PSO are described in the following section.

6.1 Particle Swarm Optimization

A novel population based optimization approach, called Particle Swarm Optimization (PSO) approach, has been used. PSO was introduced first in 1995 by Eberhart and Kennedy [19]. This new approach features many advantages; it is simple, fast and can be coded in few lines. Also, its storage requirement is minimal.

Moreover, this approach is advantageous over evolutionary algorithms in more than one way. First, PSO has memory. That is, every particle remembers its best solution (local best) as well as the group best solution (global best). Another advantage of PSO is that the initial population of the PSO is maintained, and so there is no need for applying operators to the population, a process which is time- and memory-storage-consuming. In addition, PSO is based on "constructive cooperation" between particles, in contrast with the other artificial algorithms which are based on "the survival of the fittest."[20]

PSO starts with a population of random solutions. Each particle keeps track of its coordinates in hyperspace which are associated with the fittest solution it has achieved so far. The value of the fitness (*pbest*) is stored. Another best value is also tracked, which is the global best value. The global version of the PSO keeps track of the overall best value, and its location, obtained so far by any particle in the population, which is called (*gbest*).

PSO consists of, at each step, changing the velocity of each particle toward its pbest and gbest. Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward *pbest* and *gbest*.

Several modifications have been proposed in the literature to improve the PSO algorithm speed and convergence toward the global minimum. One of the most efficient PSO versions uses a time decreasing inertia weight, which leads to a reduction in the number of iterations. The performance of this modified algorithm depends on the method of tuning the inertia weight. The most straightforward way to adjust the inertia weight is to force it to linearly decrease with the number of iterations [21]. As an alternative to inertia weight technique, constriction factors have been proposed [22]. Constriction-factor-based PSO has been proved to be superior upon inertia-weight-based PSO in terms of the number of iterations to reach the best solution.

PSO starts with a population of random solutions "particles" in a D-dimension space. The i^{th} particle is represented by $X_i = (x_{1i}, x_{i2}, ..., x_{iD})$. Each particle keeps track of its coordinates in hyperspace which are associated with the fittest solution it has achieved so far. The value of the fitness for particle *i* (*pbest*) is also stored as $P_i = (p_{1i}, p_{i2}, ..., p_{iD})$. The global version of the PSO keeps track of the overall best value (*gbest*), and its location, obtained so far by any particle in the population.

PSO consists of, at each step, changing the velocity of each particle toward its *pbest* and *gbest* according to Eqn. (1). The velocity of particle *i* is represented as $V_i = (v_{1i}, v_{i2}, ..., v_{iD})$. Acceleration is weighted by a random term *w*, with separate random numbers being generated for acceleration toward pbest and gbest. The position of the i^{th} particle is then updated according to Eqn. (2).

$$v_{id} = w * v_{id} + c_1 * Rand(\ ) * (p_{id} - x_{id}) + c_2 * Rand(\ ) * (p_{gd} - x_{id}) \tag{1}$$

$$x_{id} = x_{id} + v_{id} \tag{2}$$

where

$p_{id} = pbest$ and $p_{gd} = gbest$

A simplified method of incorporating a constriction factor is represented in Eqn. (3), where *K* is a function of c_1 and c_2 as illustrated by Eqn. (4). Eberhart and Shi [22] experimented the performance of PSO using an inertia weight as compared with PSO performance using a constriction factor. They concluded that the best approach is to use a constriction factor while limiting the maximum velocity V_{max} to the dynamic range of the variable X_{max} on each dimension. They showed that this approach provides performance superior to any other published results in the literature.

$$v_{id} = K * [\, v_{id} + c_1 * Rand(\ ) * (p_{id} - x_{id}) + c_2 * Rand(\ ) * (p_{gd} - x_{id})], \tag{3}$$

$$K = \frac{kk}{\left| 2 - \phi - \sqrt{\varphi^2 - 4\phi} \right|};\ kk = 2,\ \phi = c_1 + c_2,\ \phi > 4\,. \tag{4}$$

6.2 Objective Function

Since the problem of selecting the best descriptors can be formulated as an optimization problem, one need to define an objective function. The objective function, in this case, is made up of the recognition rate and the number of useful descriptors. In other words, it is required to maximize the recognition rate using the minimum number of descriptors. The function of the optimization algorithm is to assign a weight w_i for every descriptor so that the objective function is optimized, where w_i belongs to [0,1]. The mathematical formulation of the objective function is defined as follows:

$$J = -H + \alpha * \min(PE), \tag{5}$$

where H is the number of hits (the number of correct matches), PE is the percentage of errors of all the training images for a given set of weights, and α is a factor that is adjusted according to the relative importance of the min(PE) term. In most of the simulations, $\alpha = 0.7$ is experienced as best case most of the time.

The first term, in the objective function Eqn. (5), makes PSO search for the weights that result in the highest recognition rate and the second term makes the PSO reach to the highest recognition with the minimum number of descriptors.

6.3 Steps of the PSO Algorithm

The PSO algorithm is described as follows:

Step 1: Define the problem space and set the boundaries, i.e. equality and inequality constraints.
Step 2: Initialize an array of particles with random positions and their associated velocities inside the problem space.
Step 3: Check if the current position is inside the problem space or not. If not, adjust the positions so as to be inside the problem space.
Step 4: Evaluate the fitness value of each particle.
Step 5: Compare the current fitness value with the particles' previous best value (*pbest[]*). If the current fitness value is better, then assign the current fitness value to *pbest[]* and assign the current coordinates to *pbestx[][d]* coordinates.
Step 6: Determine the current global minimum among particle's best position.
Step 7: If the current global minimum is better than gbest, then assign the current global minimum to *gbest[]* and assign the current coordinates to *gbestx[][d]* coordinates.
Step 8: Change the velocities according to (3.205).
Step 9: Move each particle to the new position according to (3.206) and return to **Step 3.**
Step 10: Repeat **Step 3- Step 9** until a stopping criteria is satisfied.

6.4 Test Results Using PSO

The PSO algorithm can be found in the current literature at many places. Therefore, it is not necessary to describe the steps of the PSO algorithm here. However, the reader is referred to [19-22] if interested to see the PSO algorithm details.

The proposed PSO–based approach has been implemented using a MATLAB library built by the authors. In our implementation, the inertia weight w is linearly decreasing from 0.9 to 0.4, c_1 and c_2 are selected as 2, a population of 20 particles is used, and the maximum number of iterations is 300. The search process stops whenever the maximum number of iterations is exceeded.

Several experiments have been attempted to use PSO to search for the optimum descriptor weights. These experiments are summarized in Tables 1 and 2. In these tables, "No. of FDs considered" means the number of Fourier descriptors considered in the optimization process. For example, if this number is F, the PSO is supposed to search for F weights, a single weight for a single FD, that maximize the recognition rate with the minimum number of descriptors. Hence, if only f FDs can do the job, PSO will assign nonzero weights for f FDs and it will assign zero weights for F-f FDs. As a result, the number of descriptors used in the recognition process is only f out of F.

Table 1 demonstrates the computed optimized weights for different numbers of Moment descriptors. Table 2 displays about the total number of optimized weights used for different numbers of Moment descriptors and the recognition rate achieved. One can see much better recognition results than using un-weighted Moments in Section 6. However, in the experiment using occluded images only in the training, does not help improve the recognition rate of occluded images.

Table 1 Optimized weights for different numbers of Moment descriptors

Exp. No.	**1**	**2**	**3**	**4**	**5**
No. of moments considered	3	5	7	9	11
Training set*	X	X	X	X	X
Weights obtained (%)	26.2451 5.8841e-008 1.8871e-015	0 29.6124 1.3910e-007 5.6681e-007 0	0 43.1600 2.6314e-007 1.6375e-006 0 0 0	0 39.5053 2.3726e-007 1.4896e-008 0 0 0 0 0	100.0000 5.6561e-008 4.0486e-017 0 0 0 0 0 0 0 0
alpha	1e-14	1e-14	1e-14	1e-15	8e-14

Table 2 Total number of optimized weights used for different numbers of Moment descriptors and the recognition rate

Exp. No.		**1**		**2**		**3**		**4**		**5**	
No. of moments considered		3		5		7		9		11	
Training set*		X		X		X		X		X	
No. of moments Used (i.e. no. of nonzero weights)		3		3		3		3		3	
		/60	**%**	**/60**	**%**	**/60**	**%**	**/60**	**%**	**/60**	**%**
Rec Rate	**X**	57	95	56	93.33	56	93.33	54	90	57	95
	N**	13	81.25	13	81.25	13	81.25	13	81.25	13	81.25
	O	8	13.33	13	21.67	11	18.33	7	11.67	3	5
	Ave	---	63.19	---	65.41	---	64.30	---	60.97	---	60.42

*X = transformed objects, O = occluded objects, N = noisy objects.

** Out of 16 test images.

7 Conclusion and Future Work

This work has been reported to make a practical study of the Hu's Moments to the application of Object Recognition. The implementation was done on a P-IV PC using MATLAB. The ultimate results have variations depending upon the selection of Moments and Data size. The variety similarity measures and different combinations of invariant moment features, used in the process, make a difference to the recognition rate. The results have been tested using Euclidean distance similarity measures, 11 moment invariants, and different size databases. Different combinations of these parameters implied different results.

The images that have to be recognized but failed to be recognized by most of the Moment Invariant combinations are to be analyzed further. This leads to the theory of optimization to find out appropriate features or attributes in the image that made it difficult to be recognized. The methodology of PSO has been utilized successfully for this purpose. Using PSO, to find the most suitable descriptors and to assign weights for these descriptors, improved dramatically the recognition rate using the least number of descriptors.

Here are some observations for the whole discussion in the paper: Various observations about the use of different combinations of moments, noise in the images, use of variety of similarity transformations, and presence of occlusion in the images have been made. For brevity, the reader is referred to [9]. Moreover, using PSO, to find the most suitable descriptors and to assign weights for these descriptors, improves dramatically the recognition rate using the least number of descriptors.

Acknowledgments. This work has been supported by Kuwait University, Kuwait.

References

1. Hu, M.-K.: Visual Pattern Recognition by Moment Invariants. IRE Trans. Inf. Theory IT-8, 179–187 (1962)
2. Li, Y.: Reforming the theory of invariant moments for pattern recognition. Pattern Recognition 25, 723–730 (1993)
3. Flusser, J., Suk, T.: Pattern recognition by affine moment invariants. Pattern Recognition 26(1), 167–174 (1993)
4. Hu, M.-K.: Pattern recognition by moment invariants. In: Proc. IRE (Correspondence), vol. 49, p. 1428 (1961)
5. Wong, W.H., Siu, W.C., Lam, K.M.: Generation of moment invariants and their uses for character recognition. Pattern Recognition Lett. 16, 115–123 (1995)
6. Khotanzad, Hong, Y.H.: Invariant image recognition using Zernike Moments. IEEE Transactions Pattern Analysis and Machine Intelligence 16(10), 976–986 (1994)
7. Dudani, S.A., Breeding, K.J., McGhee, R.B.: Aircraft identification by moment invariants. IEEE-T on Computers 26(1), 39–46 (1977)
8. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/SHUTLER3/
9. Sarfraz, M.: Object Recognition using Moments: Object Recognition using Moments: Some Experiments and Observations. In: Sarfraz, M., Banissi, E. (eds.) Geometric Modeling and Imaging – New Advances, pp. 189–194. IEEE Computer Society, USA (2006)
10. Zhang, J., Zhang, X., Krim, H., Walter, G.G.: Object representation and recognition in shape spaces. Pattern Recognition 36(5), 1143–1154 (2003)
11. Ansari, N., Delp, E.J.: Partial Shape Recognition: a landmark based approach. IEEE Trans. PAMI 12, 470–483 (1990)
12. Gorman, J.W., Mitchell, O.R., Kuhl, F.P.: Partial shape recognition using dynamic programming. IEEE Transactions on Pattern Analysis and Machine Intelligence 10(2) (March 1988)
13. Gorilick, L., Galun, M., Sharon, E., Basri, R., Brandt, A.: Shape representation and classification using poisson equation. In: 20th international conference on computer vision and pattern recognition, vol. II-129 (2004)
14. Avrahami, G., Pratt, V.: Sub-pixel edge detection in character digitization. In: Raster Imaging and Digital Typography II, pp. 54–64 (1991)
15. Hou, Z.J., Wei, G.W.: A new approach to edge detection. Pattern Recognition 35, 1559–1570 (2002)
16. Richard, N., Gilbert, T.: Extraction of Dominant Points by estimation of the contour fluctuations. Pattern Recognition 35, 1447–1462 (2002)
17. Gonzalez, R., Woods, R., Eddins, S.: Digital Image Processing using Matlab. Prentice Hall, Englewood Cliffs (2003)
18. Jain, R., Kasturi, R., Schunk, B.: Machine Vision. McGraw-Hill, New York (1995)
19. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. IEEE Intl. Conf. Neural Networks, November/December 1995, vol. 4, pp. 1942–1948 (1995)
20. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proc. the Sixth Intl. Symposium on Micro Machine and Human Science, MHS 1995, October 4-6, pp. 39–43 (1995)

21. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: The 1998 IEEE Intl. Conf. on Evolutionary Computation Proc. IEEE World Congress on Computational Intelligence, May 4-9, pp. 69–73 (1998)
22. Eberhart, R.C., Shi, Y.: Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the 2000 Congress on Evolutionary Computation, July 16-19, vol. 1, pp. 84–88 (2000)
23. Sarfraz, M., Al-Awami, A.T.A.: Object Recognition using Particle Swarm Optimization on Fourier Descriptors, Soft Computing in Industrial Applications. In: Saad, A., Avineri, E., Dahal, K., Sarfraz, M., Roy, R. (eds.) Soft Computing in Industrial Applications: Recent and Emerging Methods and Techniques. Advances in Soft Computing, vol. 39, pp. 19–29. Springer, Heidelberg (2007)

Perceptual Shaping in Digital Image Watermarking Using LDPC Codes and Genetic Programming

Imran Usman, Asifullah Khan, Rafiullah Chamlawi, and Tae-Sun Choi

Abstract. In this work, we present a generalized scheme for embedding watermark in digital images used for commercial aims. Genetic Programming is used to develop appropriate visual tuning functions, in accordance with Human Visual System, which cater for watermark imperceptibility-robustness trade off in the presence of a series of probable attacks. The use of low-density parity check codes for information encoding further enhances watermark robustness. Experimental results on a dataset of test images show marked improvement in robustness, when compared to the conventional approaches with the same level of visual quality. The proposed scheme is easy to implement and ensures significant robustness for watermarking a large number of small digital images.

1 Introduction

Due to widespread use of interconnected networks, recent years have witnessed a striking increase in digital multimedia representation, storage and distribution. Digital images, such as those displayed in online catalogs and auction websites, are smaller in sizes, larger in numbers, and strongly differ in intrinsic characteristics.

Furthermore, watermarked images are generally exposed to a set of probable attacks. This, indeed, has triggered an increase of multimedia content piracy. Digital watermarking is a promising technique to help protect intellectual property rights and piracy control, whereby, a *watermark* is imperceptibly incorporated in the digital data containing information about the same data [1].

A low strength watermark incorporates high imperceptibility but weak robustness and vice versa [1, 2]. On the other hand, different set of attacks are associated with distinctive watermarking applications, which pose different requirements on a watermarking scheme [2]. These conceivable attacks may include intentional and/or unintentional manipulation of the marked cover work. A universal watermarking

Imran Usman · Asifullah Khan · Rafiullah Chamlawi
Department of Computer and Information Sciences,
Pakistan Institute of Engineering and Applied Sciences, PO Nilore, Islamabad, Pakistan
e-mail: {phd0406, asif, chamlawi}@pieas.edu.pk

Asifullah Khan · Tae-Sun Choi
Department of Mechatronics, Gwangju Institute of Science and technology,
1 Oryong-Dong, Buk-Gu, Gwangju 500-712, Republic of Korea
e-mail: {asif, Tae-Sun}@gist.edu.kr

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 509–518.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

scheme that can confront all attacks and at the same time fulfill all other desirable requirements is difficult to exist [2]. Therefore, while designing a watermarking system, its intended application and thus the corresponding set of anticipated attacks is of prime importance.

A watermark is generally structured using perceptual models, which exploit the characteristics of HVS in order to attain imperceptibility. Watson's Perceptual Model (WPM) [3] is an appropriate example of such models, as it has been extensively used in image compression and digital watermarking [4, 5]. These perceptual models are devised in view of invisibility requirements alone, and thus, lack in exploiting the attack information. As a result, they are inefficient to cater, concurrently, for both robustness and imperceptibility needs of a watermarking system.

A variety of watermarking schemes are proposed in literature, including transform domain techniques and spatial domain methods which make use of a predetermined set of coefficients, mostly in the middle frequency range, to serve as a tradeoff for watermark embedding. The disadvantages of such a selection are discussed in detail by Huang *et al*. [6], who also proposed a scheme in which suitable embedding positions in a block based DCT domain watermarking are selected using Genetic Algorithm. Khan *et al.* [7] have used Genetic Programming (GP) to generate expressions that provide suitable strength of alteration in each selected DCT coefficient for watermark embedding.

Low-density parity check codes (LDPC), first proposed by Gallager [8], are a class of binary linear codes used for error correction. The concept behind these codes is to add a small amount of controlled redundancy to the data, such that the added redundancy can be used to detect and correct errors which may occur due to various legitimate and illegitimate distortions. An LDPC code can be constructed with parameters (n, k), where n represent the length of the codeword and k is the message size. Bastug and Sankur [9] have used LDPC codes in order to improve the information payload of the watermarking channels. Dikici et al. [10] used LDPC codes to quantize the host signal such as to increase the message extraction probability. We utilize the error correction capability of the LDPC codes in conjunction with GP in order to achieve higher robustness in a very hostile environment where a sequence of attacks is expected.

In this work, we propose an intelligent system capable of structuring digital watermarks for practical applications, where, not only fair imperceptibility is required, but also achieving resistance against a set of conceivable attacks is considered as a more realistic approach. We use GP and the concept of wrapper to evolve appropriate Visual Tuning Functions (VTFs) in order to find both suitable location, as well as suitable strength of alteration for watermark embedding. To enhance the robustness of the system further, LDPC codes are utilized to encode the message before it is converted into a watermark.

The rest of the work is organized as follows. Section 2 presents our proposed scheme along with a brief discussion on relevant concepts, followed by results and discussion in section 3. Conclusion is presented in section 4.

2 Proposed Scheme

The response of HVS varies with spatial frequency, luminance and color of its input. This suggests that all components of a watermark may not be equally perceptible when embedded. WPM exploits sensitivities/insensitivities of HVS for assigning a perceptual slack, $\alpha(i, j)$, to each selected coefficient of the image expressed in DCT domain. We consider these selected DCT coefficients as features for the embedding phase. At the watermark extraction phase, the same features are extracted to decode the watermark. According to Watson's perceptual function [3], the visibility threshold $T'(i, j)$, incorporating the effect of frequency as well as luminance sensitivity, is further modified into $T^*(i, j)$, to take into consideration the contrast masking effect:

$$T^*(i,j) = \max\left[T'(i,j),\ \left|T'(i,j)\right|^{1-\omega} X(i,j)^{\omega}\right], \tag{1}$$

where $X(i, j)$ is AC DCT coefficient of each block and ω has been empirically set to a value of 0.7. These allowed alterations represent the perceptual mask denoted by $\boldsymbol{\alpha}$.

GP is a directed random search technique, inspired by biological evolution, for solving optimization problems [11]. In GP, a population of individual solutions (represented by a data structure, such as a tree) is repeatedly modified generation by generation. Every candidate solution is evaluated and assigned a fitness value using an application specific, predefined *fitness function*. The best individuals in a generation are maintained, while the rest are removed and replaced by the offspring of the best individuals to make a new generation. Offspring are then produced by applying genetic operators. The whole process is repeated for subsequent generations, with scoring and selection procedure in place, until the termination criterion is met.

For representing a candidate solution (VTF) with a GP tree, suitable GP *terminal set*, and GP *function set* are defined. Therefore, in our proposed scheme the GP terminal set comprises of the current value of WPM-based perceptual mask, DC and AC DCT coefficients of 8×8 block as variable terminals. Random constants in the range [-1, 1] are used as constant terminals. The GP function set comprises of simple functions including four binary floating arithmetic operators (+, -, *, and protected division), *LOG, EXP, SIN, COS, MAX and MIN*.

The functional dependency of the VTF on the characteristics of HVS and the distortion incurred by a single attack can be represented as:

$$\alpha_{VTF}(k_1, k_2) = f\left(X_{0,0}, X(i,j), \alpha(k_1, k_2), \varphi_1\right), \tag{2}$$

where i, j are block indices and k_1, k_2 are general 2-D indices. $\alpha(k_1, k_2)$ represents Watson's perceptual mask incorporating HVS dependency based on frequency sensitivity, luminance sensitivity and contrast masking effect. It is to be noted that

$\alpha_{VTF}(k_1, k_2)$ is just a functional dependency whose realization appears in equation 7. $X_{0,0}$ is the DC DCT coefficient, $X(i,\ j)$ is the AC DCT coefficient of the current block and φ_1 denote the information pertaining to the distortions caused by a single attack. However, in order to incorporate the knowledge regarding a sequence of attacks, we modify equation 2 as:

$$\alpha_{VTF}(k_1, k_2) = f\left(X_{0,0}, X(i,j), \alpha(k_1, k_2), \varphi_3\right), \tag{3}$$

where $\varphi_3 = f(\varphi_2(\varphi_1))$ represents a sequence of distortions caused by three attacks. It is to be noted that the order of the attacks is also important. Nevertheless, we have observed that in order to resist a sequence of attacks, deciding only the strength of watermarking is not sufficient. Therefore, we modify equation 3 by exploiting frequency band selection as well:

$$\alpha_{VTF}(k_1, k_2) = f\left(X_{0,0}, X(i,j), \alpha(k_1, k_2), i, j, \varphi_3\right), \tag{4}$$

where i, *j* are provided to the GP module in order to exploit position information inside a block. The attack information is implicitly learned through the fitness function defined as follows:

$$Fitness = \lambda_1 * F_i + \lambda_2 * BCR_{\varphi}, \tag{5}$$

where λ_1 and λ_2 are the corresponding weights. To attain a maximum fitness value of 1, we choose their values such that $\lambda_1 + \lambda_2 = 1$. The choice of λ_1 and λ_2 is not easy. As a rule of thumb, for applications requiring more robustness, some sacrifice can be made in terms of imperceptibility, i.e. λ_2 should be given more weightage. On the other hand, for those requiring high imperceptibility, λ_1 should be increased. In our experiments, we choose equal values of 0.5 for each of them. BCR_{φ} is the robustness measure of the watermarked work and represents Bit Correct Ratio after the set of attacks are carried out in a sequence. More information on BCR_{φ} can be found in [7]. F_i is the visual quality objective of the fitness function and consists of structural similarity index measure (*SSIM*) [12] and weighted peak signal-to-noise ratio (*wPSNR*) [13] at a certain level of estimated robustness. It is given as: $F_i = \gamma_1 * SSIM + \gamma_2 * wPSNR / 46$, where γ_1 and γ_2 are corresponding weights. The value of 46 is used empirically such as to scale the *wPSNR* value closer to 1. Since both *SSIM* and *wPSNR* considerably contribute in estimating the imperceptibility, therefore, γ_1 and γ_2 are assigned a value of 0.5 each. In this way, F_i can attain a maximum value of 1.

For enhancing the automatic selection of suitable DCT coefficient positions, we use the concept of wrapper in our GP simulation. Therefore, equation 4 is modified as follows:

$$\hat{\alpha}_{VTF}(k_1,k_2)=\left\{\begin{array}{ll}\alpha_{VTF}(k_1,k_2) & if\ \alpha_{VTF}(k_1,k_2)\geq 0\\ 0 & otherwise\end{array}\right\} \tag{6}$$

In initial generation of our GP based proposed scheme, a population of candidate VTFs is created. Each candidate VTF is evaluated and scored against the fitness function (equation 5) using n images. In our simulation, we take $n = 5$. The mean fitness value is taken as the fitness corresponding to that candidate. In this way, the whole population of candidate VTFs is scored and the best individuals are selected to be parents, so as to form the next generation by going through the genetic operations of crossover, mutation and replication. Hence, the solution space is refined generation by generation and converges to an optimum/ near optimum VTF, which is saved and used in the testing phase to watermark a dataset of m test images.

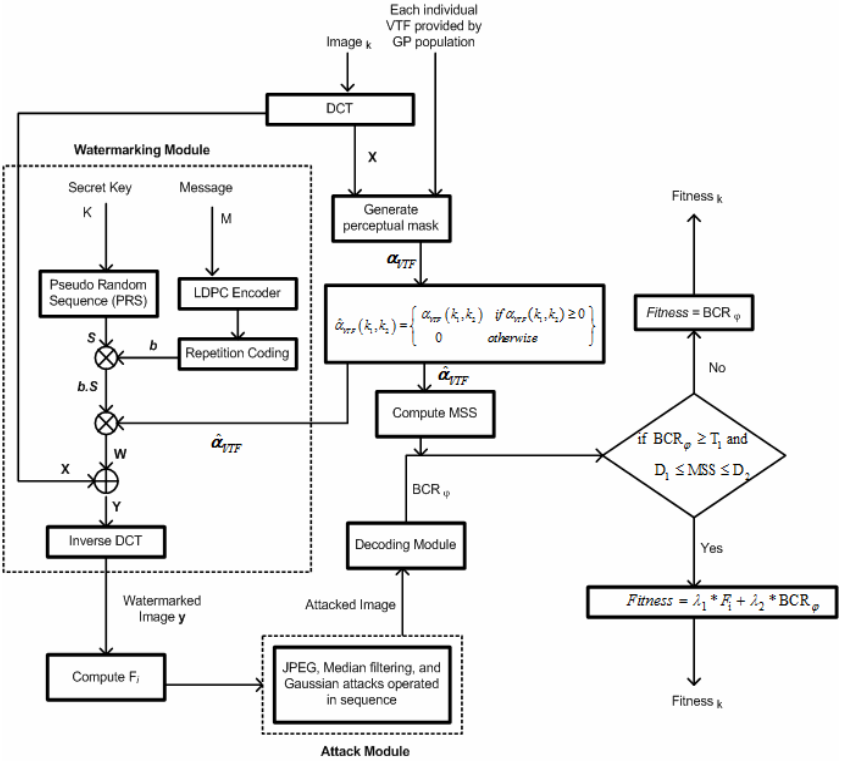


Fig. 1 Detailed representation of encoding, decoding and fitness evaluation processes of the proposed scheme

Figure 1 presents an elaborated diagram of the underlying details associated with different sub modules in encoding, decoding the fitness evaluation processes. After a candidate VTF is produced, an image is watermarked using a spread spectrum based watermarking technique such as that proposed by Hernandez *et al.* [4], and the perceptual mask derived from the candidate expression. The message bits are encoded by an

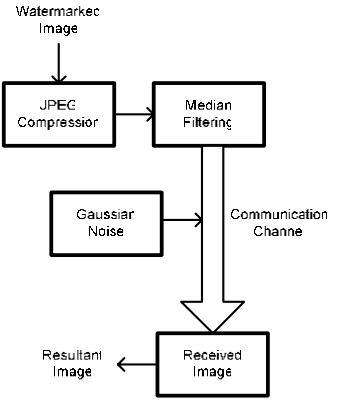


Fig. 2 Block diagram of an example attack sequence

LDPC encoder before using repetition coding. Next, the *SSIM* and *wPSNR* values are computed from the watermarked image in order to obtain F_i before presenting it to the attack module. The attack module consists of a series of attacks applied on the watermarked work. Due to space limitations, we have considered only a single attack scenario in our present work, which is presented as follows:

An Example Attack Sequence: In this attack sequence, a watermarked image is first compressed to reduce its size. Later, the owner noticed small impulse noise in the produced work and performed median filtering to reduce it, and transmitted it via a channel characterized by Gaussian noise. On the receiver side, the received watermarked image has suffered through a sequence of unintentional attacks. This attack scenario is presented in figure 2. The generalization ability of the proposed scheme is not limited to this attack scenario. It is practically viable on attack (non-geometric attack) sequences consisting of any combination of intentional and/or unintentional manipulations of the cover work. This is because the proposed method is able to exploit the hidden dependencies in HVS, which are otherwise, impossible to model using conventional approaches. Furthermore, through genetic evolution, it is able to learn the appropriate DCT bands to embed the watermark in order to make it resistant against different attack sequences.

After the series of attacks are carried out, the decoding module decodes the watermark and estimates the hidden message bits in order to compute BCR_φ. The decoding module implements the decoder proposed by Hernandez *et al.* [4].

We assume watermark power to depict robustness at the embedding stage of GP simulation. For this purpose, we use Mean Squared Strength (*MSS*) as used in [7] to estimate robustness of individuals in a GP population.

If *MSS* and BCR_φ values lie within and under certain bounds respectively, *bonus fitness* is given which includes fitness due to imperceptibility as well. Else, the fitness equals BCR_φ value. This bounding helps in selecting individuals performing well on both of the objectives, i.e. individuals that provide sufficient watermark strength without violating the imperceptibility requirement.

3 Experimental Results

The proposed scheme is implemented in Matlab. For GP simulation, we have used Matlab based GPlab toolbox [14]. In the training phase *Lena*, *couple*, *boat*, *baboon*, and *trees* images of size 128×128 are used to evolve optimal VTFs. The values of D_1, D_2, and T_1 are empirically set to 1, 220 and 0.975 respectively. The weights $\lambda_1, \lambda_2, \gamma_1$, and γ_2 are set to 0.5 each. The evolved VTF for the example attack sequence is presented in prefix notation as:

$$\begin{aligned}\hat{\alpha}_{VTF}(k_1,k_2) = {} & *(*(*(X_{0,0}, \alpha(k_1,k_2)), \cos(\log(Max(\log(i)), \log(Max)(*(X_{0,0}, \alpha(k_1,k_2)),\\ & \sin(\sin 0.674))))))), Max(\log(\log(Max(\log(Max(\log(\alpha(k_1,k_2)),\\ & \sin(0.674))), \sin(0.674)))), \sin(0.674))).\end{aligned} \tag{7}$$

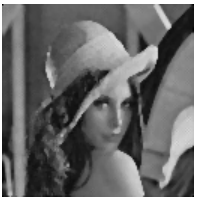
Figure 3 (a) and (b) present original and watermarked images respectively. Figure 3(c)-(h) demonstrates the severity of distortions introduced by a series of attacks in the watermarked image. Figure 3(d), (f) and (h) show the scaled difference image after each subsequent attack is applied. It is observable that most of the distortions occur at the edges, where conventional watermarking techniques usually perform the embedding. Due to intelligent position and strength selection, and employment of LDPC coding, our watermarking scheme is able to counter such a hostile attack environment.



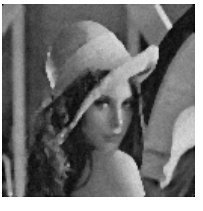
(a) Original Lena Image.



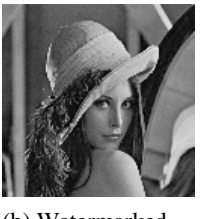
(c) JPEG attacked Lena Image.



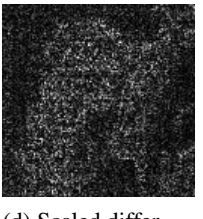
(e) (JPEG + median filtering) attacked Lena image.



(g) (JPEG + median filtering + Gaussian) attacked Lena image.



(b) Watermarked Lena image.



(d) Scaled difference image between (b) and (c).



(f) Scaled difference image between (b) and (e).



(h) Scaled difference image between (b) and (g).

Fig. 3 Severity of distortions caused by the sequence of attacks on watermarked Lena image

Table 1 shows the results obtained using the aforementioned evolved VTF for the example attack sequence. We compare our results with Hernandez et al.'s watermarking scheme [4] which make use of WPM ($\alpha(k_1, k_2)$) instead of genetically evolved VTF ($\hat{\alpha}_{VTF}(k_1, k_2)$). Table 2 shows results obtained using WPM. Comparing it to table 1, we observe that the evolved expression outperforms WPM in terms of BCR, keeping almost the same level of SSIM, and hence the same imperceptibility. For this purpose, we use some scaling factor which scales the *SSIM* value in case of WPM to that derived using the evolved VTF.

Table 1 Performance measures of the evolved VTF on 5 training images against the example attack sequence

Image Name	MSS	Mean Squared Error	Document to Watermark Ratio	wPSNR	SSIM	F_i	BCR
Lena	12.99	12.98	29.88	39.87	0.967	0.917	1
Couple	18.66	18.64	29.50	39.95	0.967	0.918	0.937
Boat	25.12	25.07	29.27	41.04	0.913	0.902	1
Baboon	18.13	18.11	29.93	41.21	0.976	0.936	0.937
Trees	19.93	19.75	30.03	38.39	0.943	0.888	1

Figure 4 shows the performance of the evolved VTF against the example attack sequence for 300 test images. These comprise of images from everyday life containing both indoor and outdoor images in order to demonstrate the generalization ability of the proposed scheme. For comparing it with Watson's perceptual shaping function, the imperceptibility in terms of SSIM is kept the same for both the methods. The effectiveness of the proposed scheme and its generalization capability to be used in order to watermark a large number of small images can be observed from the marked improvement in terms of robustness with no compromise on imperceptibility.

Table 2 Performance measures of WPM on training images against the example attack sequence. For comparison purpose *SSIM* is kept the same as in table 1 using some scaling factor

Image Name	MSS	Mean Squared Error	Document to Watermark Ratio	wPSNR	SSIM	F_i	BCR
Lena	17.09	17.07	28.69	39.75	0.964	0.914	0.75
Couple	20.68	20.61	29.02	39.92	0.967	0.917	0.937
Boat	10.14	10.09	33.22	41.89	0.915	0.913	0.937
Baboon	18.59	18.52	29.83	42.23	0.974	0.946	0.875
Trees	6.29	6.26	35.02	44.99	0.947	0.962	0.625

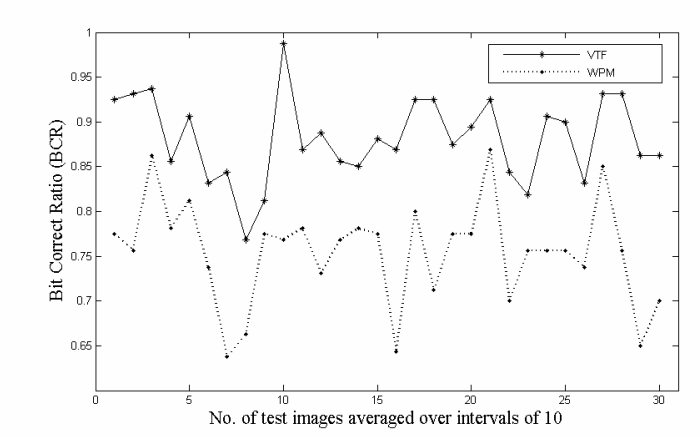


Fig. 4 BCR for the evolved VTF and WPM using 300 test images against the example attack sequence. Note that the number of images is averaged over intervals of 10

4 Conclusion

The proposed scheme presents a method of shaping a watermark in terms of robustness-imperceptibility trade off and in view of a set of probable attacks. LDPC coding is employed to further increase its robustness. In real world applications, a watermarked work is vulnerable to such circumstances. Experimental results characterize that the proposed approach outperforms the conventional Watson's perceptual model for watermarking a large number of small images. This improvement is achieved by exploiting both the positions and strength of alteration in DCT coefficients and by making use of error correction strategy at the encoding stage.

Acknowledgements

The financial support provided by HEC, Pakistan, under PhD scholarship program No. 17-5-1(Cu-204) HEC/Sch/2004 is highly acknowledged.

References

1. Cox, I.J., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann, San Francisco (2002)
2. Huang, H.-C., Jain, L.C., Pan, J.-S.: Intelligent watermarking techniques. World Scientific Pub. Co. Inc., Singapore (2004)
3. Watson, B.: Visual optimization of DCT quantization matrices for individual images. In: Proc. AIAA Computing in Aerospace 9, San Diego, CA, pp. 286–291 (1993)
4. Hernandez, J.R., Amado, M., Perez-Gonzalez, F.: DCT-Domain watermarking techniques for still images: Detector performance analysis and a new structure. IEEE Trans. on Image Processing 9(1), 55–68 (2000)

5. Li, Q., Cox, I.J.: Using Perceptual Models to Improve Fidelity and Provide Invariance to Valumetric Scaling for Quantization Index Modulation Watermarking. In: ICASSP, vol. 2, pp. 1–4 (2005)
6. Shieh, C.S., Huang, H.C., Wang, F.H., Pan, J.S.: Genetic watermarking based on transform domain techniques. Pattern Recognition 37(3), 555–565 (2004)
7. Khan, A., Mirza, A.M.: Genetic perceptual shaping: Utilizing cover image and conceivable attack information during watermark embedding. Information Fusion 8(4), 354–365 (2007)
8. Gallager, R.G.: Low-density parity-check codes. IRE Trans. Inform. Theory 8, 21–28 (1962)
9. Bastug, A., Sankur, B.: Improving the payload of watermarking channels via LDPC coding. IEEE Signal Processing Letters 11(2) (2004)
10. Dikici, C., Idrissi, K., Baskurt, A.: Dirty-paper writing based on LDPC codes for data hiding. In: Gunsel, B., Jain, A.K., Tekalp, A.M., Sankur, B. (eds.) MRCS 2006. LNCS, vol. 4105, pp. 114–120. Springer, Heidelberg (2006)
11. Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D.: Genetic programming an introduction: On the automatic evolution of computer programs and its applications. Morgan Kaufmanns Publishers, Inc., San Francisco (1998)
12. Wang, Z., Bovik, A.C., Sheikh, H.R.: Image quality assessment: From error visibility to structure similarity. IEEE Trans. on Image Processing 13(4), 600–612 (2004)
13. Voloshynovskiy, S., Herrigel, A., Baumgaetner, N., Pun, T.: A stochastic approach to content adaptive digital image watermarking. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 211–236. Springer, Heidelberg (2000)
14. GPlab toolbox and user's manual, `http://gplab.sourceforge.net/download.html`

Voice Conversion by Mapping the Spectral and Prosodic Features Using Support Vector Machine

Rabul Hussain Laskar, Fazal Ahmed Talukdar, Rajib Bhattacharjee, and Saugat Das

Abstract. This paper presents an alternative voice conversion technique using support vector machine (SVM)-regression as a tool which converts a source speaker's voice to specific standard target speaker. The main objective of the work is to capture a nonlinear mapping function between the parameters for the acoustic features of the two speakers. Line spectral frequencies (LSFs) have been used as features to represent the vocal tract characteristics. We use kernel induced feature space with radial basis function network (RBFN) type SVM that uses gaussian kernel. The intonation characteristics (pitch contour) is modified using the baseline technique, i.e. gaussian normalization. The transformed LSFs along with the modified pitch contour are used to synthesize the speech signal for the desired target speaker. The target speaker's speech signal is synthesized and evaluated using both the subjective and the listening tests. The results signify that the proposed model improves the voice conversion performance in terms of capturing the speaker's identity as compared to our previous approach. In the previous approach we used feed forward neural network (FFNN) based model for vocal tract modification and codebook based method for pitch contour modification. However, the performance of the proposed system

Rabul Hussain Laskar
NIT Silchar-10, Assam, India
e-mail: laskar_r@nits.ac.in

Fazal Ahmed Talukdar
NIT Silchar-10, Assam, India
e-mail: fazal@nits.ac.in

Rajib Bhattacharjee
NIT Silchar-10, Assam, India
e-mail: rajib13_nits@yahoo.co.in

Saugat Das
NIT Silchar-10, Assam, India
e-mail: saugat_nits@yahoo.co.in

J. Mehnen et al. (Eds.): Applications of Soft Computing, AISC 58, pp. 519–528.
springerlink.com © Springer-Verlag Berlin Heidelberg 2009

can further be improved by suitably modifying various user-defined parameters used in regression analysis and using more training LSF vectors in the training stage.

1 Introduction

The phenomenon of voice conversion [1]-[4] is to modify a source speaker utterance, as if spoken by a specified target speaker. The main aim of voice conversion is to design a system that can modify the speaker specific characteristics of the source speaker keeping the information and the environmental conditions, contained in the speech signal intact. In our day-to-day life, individuality in one's voice is one of the most important aspects of human speech. Our main objective in the proposed support vector machine [5]-[11] model is to design and develop a voice conversion technique which helps in maintaining the speaker's individuality or speaker's identity [23][24] in the synthesized speech. Various potential applications of voice conversion are- customization of Text-To-Speech System, developing speaker recognition and speaker verification systems in security and forensic applications, movie dubbing, animation, karaoke etc., hence the motivation for our work.

In the area of speech signal processing, isolating the characteristics of speech and speaker from the signal is a challenging problem. Mainly, two problems are to be addressed for the proposed voice conversion technique:

(a). As it is not possible to modify all the speaker specific characteristics, vocal tract characteristics have been chosen as it is of significance importance and among the prosodic features pitch contour has been taken.

(b). Define a mapping function between the source and the target features such that the transformed speech signal mimics the target speaker's voice.

The vocal tract characteristics are represented by various acoustic features, such as formant frequencies, formant bandwidths, spectral tilt, linear prediction coefficients (LPCs)[1], cepstral coefficients [1], reflection coefficients (RCs), log area ratios (LARs). For mapping the speaker-specific features between source and target speakers, various models have been explored in the literature. These models are specific to the kind of features used for mapping. For instance, gaussian mixture models (GMMs) [4][17] and vector quantization (VQ) [19][20], fuzzy vector quantization (FVQ), linear multivariate regression (LMR), dynamic frequency warping (DFW), radial basis function networks (RBFNs) [12]-[14], feed forward neural network [27] are widely used for mapping the vocal tract characteristics. Linear models, scatter plots technique [16], GMMs [15], sentence contour codebook [16] and segmental pitch contour model [1][20] are the techniques used for mapping the pitch contour. In this work, support vector machine based model has been used for mapping vocal tract characteristics. To capture the local dynamics, pitch contour has been chosen which has been modified using the base line gaussian normalization technique.

The rest of the paper is organized as follows. In section 2, the proposed voice conversion system is discussed. In section 3, 4 & 5, mapping of the vocal tract and

the pitch contour and their integration have been described. In section 6, the testing of the model and the results has been dealt with. The whole idea is summarized in section 7 and the area for future work is specified in Section 8.

2 Proposed Voice Conversion System

The block diagram for the proposed voice conversion system is shown in Fig. 1. In our work, we have proposed an algorithm to apply the support vector machine for mapping the vocal tract characteristics. The vocal tract characteristics of a particular speaker is nonlinear in nature. Moreover, the most significant information about the speaker identity lies in the vocal tract characteristics. Therefore, for voice conversion, it needs to capture a nonlinear mapping function between the vocal tract characteristics of the source and the target speaker. The main reason for exploring the SVM for mapping the vocal tract characteristics is that it captures a nonlinear mapping function between the features space of the source and the target. Unlike the back-propagation algorithm, kernel based SVM follow the structural risk minimization problem and operates only in batch mode. The SVM is a learning machine that is based on statistical learning theory, and is different from either parametric models (like GMM) or non-parametric models (like FFNN) which minimizes the upper bound on the expected test error. In our work, SVM with gaussian kernel to RBFN is used as it best fits on the data when number of data is more. The use of RBFN kernel basically gives an alternative solution by projecting the data into high dimensional acoustic feature space and a nonlinear mapping to that feature space can be implicitly performed without increasing the number of tunable parameters. For RBFN, the number of radial basis functions, their centers, linear weights and basis levels all are computed automatically. It has been observed that SVM can be used as a classifier to identify the speakers [7]. The SVM classifier used for digit recognition problem also give a good performance [6]. Hence, the support vector machine based model has been proposed for mapping the vocal tract characteristics from the source to the target speaker.

For mapping the intonation patterns between a speaker pair, there exist different methods with varying degree of complexity. There is an error between the mapped pitch contour [15][16] and the target pitch contour with respect to local rises-falls in the intonation patterns. These rises-falls characterize the stress patterns present in the utterance which are basically depended on the nature of the syllable and the linguistic context associated to the syllable. To capture the local variations to certain extent gaussian normalization is used at the word level, that are manually extracted from the sentences of both the source and the target speakers. The sentences used for preparing the database are rich in intonation patterns and are recorded for Indian speakers. Most of the words in the database are monosyllable, disyllable or trisyllable. In Indian context, the syllable carries the most significant information about the intonation patterns and co-articulation affects. Therefore, it is expected that gaussian normalization used at word level, for modification of pitch contour, can give a better performance by capturing the local variations of the target pitch contour.

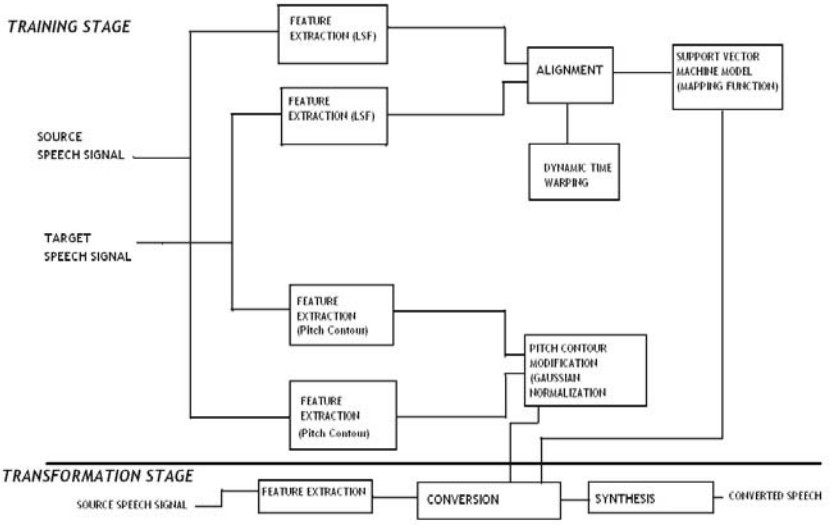


Fig. 1 Proposed voice conversion system

3 Mapping the Spectral Characteristics

3.1 Features Extraction and Time Alignment

The basic shape of the vocal tract can be characterized by the gross envelope of linear prediction (LP) spectrum. LP spectrum can be roughly represented by a set of resonant frequencies and their bandwidths [22][23]. LPC parameters are obtained using the LP analysis. In our work, another parameter set LSFs derived from LPCs is used to describe the vocal tract characteristics, as it possess good interpolation property [1]-[3]. For deriving the mapping function using proposed technique, the system has to be trained with LSFs extracted from the source and the target speaker's speech signal. For this, we prepared the text transcription for 100 English sentences. The sentences are of Indian context. These sentences are recorded for 2 male and 2 female speakers maintaining a good signal-to-noise ratio. Also all speakers are adult and have a normal utterance. The duration of sentence is varying between 3-5 seconds. The specifications of the recorded sentences are 8 KHz sampling frequency and 16 bit resolution and are recorded using mono-channel. To capture the relationship between the vocal tract shapes of the source and the target speakers, it needs to associate the time aligned vocal tract acoustic features [19] of the source and the target speakers. Dynamic time warping (DTW) [17][19] algorithm is used to derive the time aligned vocal tract acoustic features. Thus the database for both the source and the target speakers are prepared which consists of time-aligned LSFs. Approximately, 40,000 LSF vectors have been obtained. 80% of the database is used for training the system and the remaining 20% are used in cross-validation.

3.2 Training and Testing the Proposed System

The training of the features is done through support vector regression. For a given set of input and output vectors, the goal of regression is to fit a mapping function which approximates the relation between the data points and is used later to predict the output of a given new input feature vectors. The prepared database for training, for both the source and the target, contains around 40,000 LSF vectors in a 10 dimensional feature space. But 3000 to 5000 vectors are used for training the system and the performance is studied. A large number of data gives a better performance at the cost computational complexity. As the SVM operates in batch mode, all the features for both the source and the target are fed to the machine for machine learning and for establishing a mapping between the source and the target acoustic feature spaces. In our work, a nonlinear regressive model which is approximated by a scalar-valued nonlinear function is used. So an algorithm is proposed for deriving the mapping function. The procedure for training the machine, i.e., obtaining a mapping function between the source speaker's LSFs and target speaker's LSFs is given in Table-1 and Table-2 shows the overall procedure for the transformation of source speaker's LSFs to target speaker's LSFs.

Table 1 Steps for deriving the weight matrix and bias

1. 3000 LSF vectors, which are scattered in a 10-dimensional feature space,for both the source and the target are taken. So, size of LSFs vectors will be [3000,10] such that **X**[3000,10] and **Y**[3000,10] denote the source and the target LSFs respectively.
2. Now as scalar-valued nonlinear function is used, a mapping function is determined between **X**[3000,10] and Y[3000,1] in terms of a weight vector as W1[3000,1], i.e., a mapping is obtained between the source LSFs and the first column of the target LSFs. The vector W1 is the difference of Lagrange multipliers for e-insensitive loss function with interpolation error e=0.001 and the upper bound C=100.
3. The step 2 is repeated 10 times and in each iteration a new column of **Y**[3000,10] is fed and the weights are stored in matrix such that size of **W** is [3000,10].
4. Along with the weight matrix another scalar parameter bias calculated implicitly in the regression time and is stored in an array. The value of the bias is zero for RBF kernel. For other type of kernels or loss functions, the bias needs to be calculated either from the average of support vectors with interpolation error (e) or from the output training vectors, kernel matrix and weight vectors.
5. Thus a mapping function is obtained, in terms of two parameters weight and bias, which are used in the transformation stage.

4 Mapping the Prosodic Features

Once the mapped LSFs are obtained, the pitch contour is modified as it carries significant information about the speaker identity and also for improving the naturalness of the synthesized output [21][25]. Pitch contour is one of the most significant speaker-dependent prosody and is defined as the rises and falls of the fundamental frequency over time [24]. To modify one speaker's pitch contour to another speaker's pitch contour, a linear transformation function is used on a frame

Table 2 Steps for transforming LSFs

1. A new set of LSF vectors i.e. testing data corresponding to the utterance of the source speaker is taken.The size of the testing data is around **S**[300, 10] corresponding to a particular sentence of the source.
2. The LSFs of the source speaker, used in training are taken.
3. Now a kernel(**H**) matrix is formed from the training and testing data **X**[3000,10] and **S**[300,10] respectively, such that size of the kernel matrix becomes **H**[3000,300].
4. The kernel matrix and the weight matrix, derived in the training stage, are combined to get the mapped LSFs corresponding to the utterance of the target speaker and is called tested data for the target.
5. To get the final mapped LSFs of the target speaker, the bias term is added to the mapped LSFs obtained in step 4.

by frame basis based on the average ratio of each of the speakers. The pitch contour is extracted using the knowledge of instants of significant excitation, also known as epochs that are basically the instants of glottal closures in voiced regions and some random excitation in unvoiced regions. The time interval between the two epochs in the voiced portion of the speech signal corresponds to the pitch period. The pitch contour is obtained from the epoch interval plot corresponding to voiced segments. The pitch contours are extracted at the word level from the parallel words uttered by the source and the target speakers. But as most of the words are mono, di or tri-syllable(s), so it can capture the local variations to certain extent. The mean and the variances of the source and the target speakers for each of the words are calculated. The average values of the mean and the variances are obtained and from which the slope and the bias are calculated. This slope and bias parameters are used to fit a linear mapping function using gaussian normalization to map the target pitch contours corresponding to voiced segments.

5 Synthesis of Speech Signal Using Modified Features

Modified LP residual signal for the target speaker is obtained by deriving the new epoch interval plot from the mapped pitch contour. The epoch sequences are obtained from the new epoch interval plot. The original epoch sequences in the voiced segments of the source speaker are replaced by new epoch sequences and epoch sequences corresponding to unvoiced segments of the original source speaker are left unchanged. Thus a new sequence of epochs of the source speaker's original LP residual signal is obtained. The rest of the excitation signal between the two successive original epochs is resampled, and copied between the successive new epochs associated with them. This gives the modified LP residual for the desired target speaker.

To combine the effect of both the mapped pitch contour and vocal tract characteristics, the modified LP residual is used to excite the time-varying vocal tract filter. The vocal tract filter is represented by the modified LPCs derived from the predicted LSFs using SVM regression tool. The convolved signal between the

modified LP residual and the modified vocal tract filter is the synthesized output speech signal for the desired target speaker.

6 Performance Evaluation of the Proposed System

The LSFs mapped using the proposed SVM model along with the desired target speaker's LSFs are shown in the Fig. 2 for a particular frame of LSFs. Along the X-axis is the order of the LSFs and along the Y-axis is the magnitude of the LSFs. It is observed that the mapped LSFs closely follows the desired target LSFs. There are some predicted LSFs that deviates from their desired target values. These may be due insufficient training vectors used during the training phase. We have tested the proposed system for around 8000 test LSF vectors, kept in the database for cross-validation. The results in all the cases shows almost simillar performance. It is also observed in all the cases, that the lower order predicted LSFs closely the follows the desired target speaker's LSFs. There are some deviations that takes place for higher order LSFs. As lower order LSFs are more significant as compared to the higher order LSFs, so some of the high frequency spectral distortions that take place are not severe. These spectral distortions do affect the quality of the synthesized signal. It is also observed that the predicted values of the LSFs lies within 0-Π range, therefore the predicted vocal tract filter derived from the mapped LSFs are always stable.

The mapped pitch contour for male to female conversion indicate that, the mapped pitch contour captures the global behavior of the target pitch contour. It is observed that the gaussian normalization applied at sentence level can not capture the local dynamics present in the pitch contour. It only maps the average behavior of the target pitch contour. So in our work, gaussian normalization is

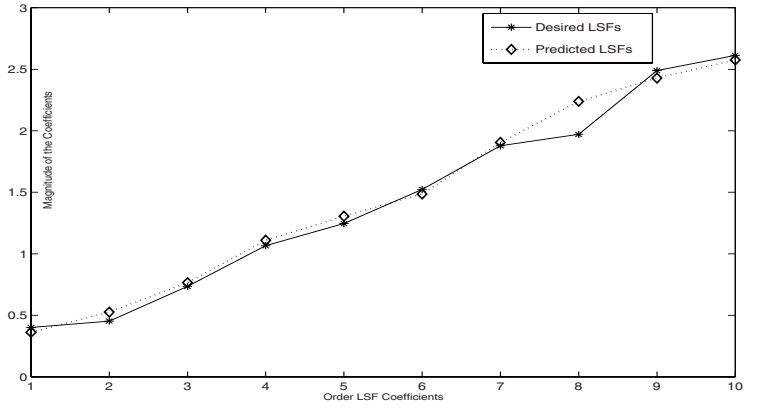


Fig. 2 The desired and the mapped LSFs of the target speaker using the proposed model for male to female conversion

used at word level and it is observed that it captures the local variations to certain extent along with the global variations. It is also observed that there exist certain jumps in the mapped pitch contour at the point of discontinuity. If there is a sudden jump in the original source pitch contour, the gaussian normalization method fails to capture this jump in the mapped pitch contour. These types of distortions may lead to spectral discontinuities in the synthesized target speech signal.

The performance of the mapping functions can be evaluated by using subjective and objective measures. In this work, the basic goal is the voice conversion, therefore the mapping functions are evaluated by using perceptual tests i.e., listening tests. In this work, one separate mapping function has been developed for transforming the speaker voices: - Male to Female (M-F), Female to Male(F-M) For this case, five utterances have been synthesized using their associated mapping functions. Listening tests are conducted to assess the desired (target) speaker characteristics present in the synthesized speech. The recorded speech utterances of the target speaker and the corresponding synthesized speech utterances had been made available to listeners to judge the relative performance. The listeners were asked to give their opinion score on a 5-point scale. The rating 5 indicates the excellent match between the original target speaker speech and the synthesized speech. The rating 1 indicates very poor match between the original and the synthesized speech and the other rating indicate different levels of deviation between 1 and 5. The obtained MOS shown in Table 3, indicate that the transformation is effective if source and target speakers are from different genders. The MOS shows that the performance of the proposed method is comparable to our previous method. In the previous method, feed forward neural network based model was used for mapping the vocal tract characteristics and codebook based model for mapping the pitch contour for voice conversion [27].

Table 3 MOS for performance evaluation of the integrated systems

Speaker combination	Using proposed SVM based integrated method	Using FFNN based integrated method
M-F	3.86	3.61
F-M	3.99	3.94

7 Summary and Conclusion

In this paper, the support vector regression tool has been used for modifying the vocal tract characteristics between the source and the target speakers. The result shows that the mapped LSFs follows the desired target LSFs closely and the synthesized speech signal contains more target speaker's identity. In the present work, LSFs derived from LPCs were used for representing the shape of the vocal tract. The target speaker's speech was synthesized using the parameters derived from the mapping functions correspond to vocal tract system and prosodic characteristics. From

the perceptual tests, it was found that the voice conversion is more effective, if the source and the target speakers belong to different genders. Subjective evaluation also indicate that the developed voice conversion system has improved performance. We are trying to improve the performance of the system, by training the system more efficiently with large number of training data and also suitably selecting the different user defined parameters.

8 Future Work

The actual implementation of support vector regression involves the use of various user defined parameters. So, the performance of the machine mostly depends on the choice of these parameters. In this paper, the mapping functions corresponding to source characteristics (shape of the glottal pulse) has not been included. The duration patterns and the energy profiles between the source speaker and the target speaker are not also considered. Developing proper mapping functions to transform the duration patterns and energy profiles may improve the overall performance of the system.

References

1. Turk, O., Arslan, L.M.: New methods for vocal tract and pitch contour transformation. In: EUROSPEECH, Geneva, Switzerland (2003)
2. Kain, A.: High resolution voice conversion, PhD Thesis, Oregon Health & Science University (October 2001)
3. Barrobes, H.D.: Voice conversion applied to Text-To-Speech systems, PhD Thesis, University Politecnica de Catalunya, Barcelona (May 2006)
4. Stylianou, Y., Cappe, Y., Moulines, E.: Continuous probabilistic transform for voice conversion. IEEE Trans. on Speech and Audio Processing 6(2), 131–142 (1998)
5. Burges, C.: A tutorial on support vector machine for pattern recognition. In: Fayyad, U. (ed.) Proc. of Data Mining and Knowl. Discovery, pp. 1–43 (1998)
6. Scholkopf, B., Sung, K.K., Burges, C., et al.: Comparing support vector machine with gaussian kernels to radial basis function classifiers. IEEE Trans. on Signal Processing 45(11), 1–8 (1997)
7. Schmidt, M.S.: Identifying speaker with support vector network. In: Proc. of 28th Symposium on the Interface (INTERFACE 1996), Sydney, Australia (1996)
8. Smola, A.J., Scholkopf, B.: A tutorial on support vector regression, NeuroCOLT2 Technical Report Series, NC2-TR-1998-030, pp. 1–66 (1998)
9. Vapnik, V.: Statistical Learning Theory. John Wiley and Sons, Inc., New-York (1998)
10. Francis, E.H.T., Cao, L.J.: Modified support vector machine in financial time series forecasting. Neurocomputing 48, 847–861 (2002)
11. Scholkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond, 1st edn. MIT Press, Cambridge (2001)
12. Bors, A.G.: Introduction of the radial basis function networks. Online Symposium for Electronics Engineers, DSP Algorithms: Multimedia 1(1), 1–7 (2001)
13. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River (1998)

14. Iwahashi, N., Sagisaka, Y.: Speech spectrum conversion based on speaker interpolation and multi-functional representation with weighting by radial basis function networks. Speech Communication 16, 139–151 (1995)
15. Inanoglu, Z.: Transforming pitch in a voice conversion framework, M. Phil Thesis, St. Edmund's College, University of Cambridge (2003)
16. Chappell, D.T., Hansen, J.H.L.: Speaker specific pitch contour modeling and modification. In: Proc. of IEEE, ICASSP, pp. 885–888 (1998)
17. Lee, K.S.: Statistical approach for voice personality transformation. IEEE Trans. on Audio, Speech and Language Processing 15(2), 641–651 (2007)
18. White, G.M., Neely, R.B.: Speech recognition experiment with linear prediction, bandpass filtering and dynamic programming. IEEE Trans. on Acoustics, Speech, and Signal Processing 24(2), 183–188 (1976)
19. Abe, M., Nakamura, S., Shikano, K., Kuwabura, H.: Voice conversion through vector quantization. In: Proc. of IEEE, ICASSP, pp. 565–568 (1998)
20. Turk, O., Arslan, L.M.: Subband based voice conversion. In: Proc. of the ICSLP, USA, vol. 1, pp. 289–292 (2002)
21. Childers, D.G., Yegnanarayana, B., Wu, K.: Voice conversion: Factor responsible for quality. In: Proc. of IEEE, ICASSP, pp. 530–533 (1985)
22. Kain, A., Macon, M.: Spectral voice conversion for Text-To-Speech synthesis. In: Proc. of ICASSP, vol. 1, pp. 285–288 (1998)
23. Kuwabura, H., Sagisaka, Y.: Acoustic characteristics of speaker individuality: Control and conversion. Speech Communication 16, 165–173 (1995)
24. Akagi, M., Ienaga, T.: Speaker individuality in fundamental frequency contours and its control. Journal of Acoust. Soc. Japan 18(2), 73–80 (1997)
25. Takagi, T., Kuwabara, H.: Contribution of pitch, formant frequency and bandwidth to the perception of voice-personality. In: Proc. of ICASSP, pp. 889–892 (1986)
26. Rao, K.S., Yegnanarayana, B.: Prosody modification using instants of significant excitation. IEEE Trans. on Audio, Speech and Language Processing 14, 972–980 (2007)
27. Rao, K.S., Laskar, R.H., Koolagudi, S.G.: Voice transformation by mapping the features at syllable level. In: Ghosh, A., De, R.K., Pal, S.K. (eds.) PReMI 2007. LNCS, vol. 4815, pp. 479–486. Springer, Heidelberg (2007)

Author Index

Abraham, Ajith 23
Ahn, Chang Wook 463
Akbarzadeh-T., Mohammad-Reza 159, 463
Al-Awami, Ali Taleb Ali 499
Anitha, R. 223
Araabi, Babak N. 93, 115

Bagheri, Ahmad 83
Bahrepour, Majid 423, 463
Barai, S.V. 203
Barranco, Carlos D. 31
Basaran, M.A. 265
Bendtsen, Jan Dimon 13
Bertram, Torsten 379
Bharadwaj, K.K. 327
Bhattacharjee, Rajib 519
Bhattacharyya, Siddhartha 53
Boger, Zvi 63
Bong, David Boon Liang 483
Borges, Patrick 411
Braun, Jan 379

Campaña, Jesús R. 31
Campos, Mauro 411
Carvalho, Ana Lisse 243
Castro, Ana Karoline 243
Chamlawi, Rafiullah 509
Cheloi, Raman 423
Chien-hsun, Huang 171
Choi, Tae-Sun 453, 509
Cioppa, Antonio Della 367
Coello, Carlos Artemio Coello 463

Das, D. 193
Das, Saugat 519
Davarynejad, Mohsen 463
Davoodi, Mansoor 347
Dayjoori, Kiarash 83
De, Sourav 53
Dugošija, Djordje 149, 337
Dutta, Paramartha 53

Eftekhari, Armin 305
Egrioglu, Erol 265

Falco, Ivanoe De 367
Fernandes, Célio 357
Filipović, Vladimir 149, 337
Forouzanfar, Mohamad 305

Ganguly, Sanjib 193
Gaspar-Cunha, A. 357
Ghodsypour, Seyed Hassan 315
Ghomi, Seyed Mohamad Taghi Fatemi 315
Günther, Robert 139
Gutiérrez, Juan Manuel 73

Hakan, Aladag C. 265
Hassanien, Aboul Ella 23
Hasuike, Takashi 285
Hoffmann, Frank 379
Hosseini, S. Hadi 93, 115

Ishii, Hiroaki 285
Iswandy, Kuncup 213

Jamali, Ali 83

Karimpour, Ali 399
Khajekaramodin, Abbas 159
Khan, Asifullah 453, 509
Khanesar, Mojtaba Ahmadieh 305
Koivisto, Hannu 275
Kojić, Jelena 337
Komal 127
König, Andreas 213
Kratica, Jozef 149, 337
Krettek, Johannes 379
Krohling, Renato A. 3, 411
Kuhn, Dimitri 139
Kulkarni, Anand J. 441
Kumar, Dinesh 127, 491
Kumar, Shakti 491

Lai, Kin Keung 431
Laskar, Rabul Hussain 519
Lee, Jaewan 41
Leija, Lorenzo 73

Mahdipour, Elham 423
Mahmood, Muhammad Tariq 453
Maisto, Domenico 367
Mateo, Romeo Mark A. 41
Medina, Juan M. 31
Mehnen, Lars 233
Mężyk, Edward 295
Mohades, Ali 347
Mohammad, Tayarani 389
Moreno-Barón, Laura 73
Muñoz, Roberto 73
Murthy, Hema A. 103

Nariman-Zadeh, Nader 83
Nielsen, Jens Dalsgaard 233

Padgett, Clifford W. 181
Pai-yung, Tsai 171
Pariz, Naser 399
Peters, James F. 23
Pinheiro, Plácido Rogério 243
Plascencia, Alfredo Chávez 13
Pontes, António J. 357
Prabu, S. 103
Preindl, Bastian 233
Pulkkinen, Pietari 275

Rai, C.S. 491
Ramakrishnan, S.S. 103
Rattay, Frank 233
Reza, Akbarzadeh Toutounchi Mohammad 389
Rezaei, Jafar 347
Rigo, Daniel 3
Rowhanimanesh, Alireza 159, 399

Saad, Ashraf 181
Sahoo, N.C. 193
Sarfraz, Muhammad 499
Saroj 327
Scafuri, Umberto 367
Schaefer, Gerald 23, 473
Sedghi, Saeed 463
Shabanian, Mahdieh 93, 115
Sharma, S.P. 127
Srividhya, V. 223

Tai, K. 441
Talukdar, Fazal Ahmed 519
Tamanini, Isabelle 243
Tarantino, Ernesto 367
Thamma, Priyanka 203
Teshnehlab, Mohammad 305
Ting, Kung Chuang 483
Tošić, Dušan 149, 337

Unold, Olgierd 295
Uslu, V. Rezan 265
Usman, Imran 509

Valle, Manel del 73
Viana, Júlio C. 357
Vidhya, R. 103

Wang, Shouyang 431
Wang, Yin Chai 483
Weicker, Karsten 139

Yaghoobi, Mahdi 423
Yakhchali, Siamak Haji 315
Yolcu, Ufuk 265
Yu, Lean 431
Yuen, Kevin Kam Fung 255

Zhou, Huiyu 473

Subject Index

3D-shape recovery 453, 454, 455

active queue management 93, 115
aggregation operators 255, 256, 260, 262, 264
ANFIS 83-91, 93, 95, 97, 99, 100
artificial immune system 295, 296, 302
artificial neural networks 63, 65, 103, 203, 204, 265
association rules 224, 225, 226, 227, 230
auto-associative clustering 65, 66
Availability 127, 128, 131

backpropagation network 103
bilateral filtering 453, 454, 455, 458
bit-level 483, 486, 489
bivariate fuzzy time series 265-267, 272

cashew chestnut industrialization process 243
cement 203-205, 210
censored data 411, 412, 413, 418, 421
chemistry 181, 182, 184, 187
classification 23, 24, 26-29, 214, 215, 218
collective intelligence 441, 443
colour palette 473, 474, 479
colour quantisation 473, 474, 479, 480, 481
combinatorial optimization 345
comprehensive learning 193, 194, 201
compressive strength 203, 204, 209-211
computational intelligence 23, 24
computational tool 243
computer-aided diagnosis 23
congestion control 93-95, 100, 115-117, 119
constrained GA 159, 161, 162
contingency plan 4, 6
control input 399-401
controllability 399-403
correlation coefficients 61
coverage problem 347-350, 353
critical path analysis 323
cyclic voltammetry 74, 78, 79

data mining 295, 302
decision attitudes 255, 259, 260, 261, 262, 264
decision making 255, 264
decision rule mining 327
digital watermarking 509, 510
discrete cosine transform 509
distributed objects 41, 42, 43, 45
distributed optimization 441, 450
distribution system expansion planning 193, 194, 198
disturbance analysis 115
diversity strategy 431, 436, 437, 439
docking problem 139, 140
dummy images 487

earthquake 159, 163, 165, 167
electronic tongue 73, 74, 78

ensemble learning 431-435, 437, 438, 439
ensemble strategy 431, 432, 434, 439
evolutionary computation 216, 365, 421
extremal optimization 367, 368, 369, 375

face recognition 483, 489, 491, 492, 497
fast screening 139
fault-tolerant 213
feed forward neural networks 265
fitness approximation 463, 464, 465
flexible search 32, 36, 39, 40
focus measure 453, 454, 457, 458
forecasting 266, 267, 269, 272
function estimation 275, 276, 277
fuzzy c-means 473, 474, 475, 480, 481
fuzzy clustering 41, 43, 45
fuzzy granulation 465
fuzzy logic 4, 7, 11, 13, 17, 203, 204, 205, 295
fuzzy object-relational databases 31, 39
fuzzy random programming 285, 286, 287, 293
fuzzy time series 265, 266, 268

gaussian normalization 519, 521, 524
gene expression programming 203, 204, 205
generalized gamma family 411, 414
generalized precedence relations 315, 316, 323
Genetic algorithms 127, 128
genetic algorithms 54, 87, 140, 181, 182, 337, 345, 353, 389
genetic fuzzy systems 275, 283
genetic programming 509, 510
genetic regression 182
GENSO 234, 236, 238
geographical information system 112
grids 367, 370
group decision making 3

heavy metals 73, 74, 78, 80
hidden information detection 463
hierarchical production rules 327, 328
hub location 149, 150, 158
human visual system 509

image 499, 500, 506
independent component analysis 491, 492
information fusion 255
initialization 275, 276, 277, 281
injection molding 357, 358, 361
instance based learning 380, 388
interaction 380, 381
intonation pattern 521

knowledge discovery 327

Lambda-Tau methodology 127, 129, 130
landslide susceptibility mapping 103
linear discriminant analysis 305, 306
link quality prediction 233, 235
load balancing 41, 42, 43, 45, 48
low-density parity check codes 509, 510

machine learning 233, 237, 238, 239, 240
Maintainability 127, 128, 131, 133, 134, 136
Mamdani fuzzy models 275, 276, 283
mapping 367, 368, 369, 370, 371, 373
MCDM 3, 243
medical image segmentation 23, 24, 25
medical informatics 23, 24
mixture resolution 73
MLSONN 54, 55, 57, 58, 61
mobile robots 13, 21
moment descriptors 499, 500, 505
multi-agent learning 431
multi-agent systems 441
multi-level constrained GA 399
multi-level image segmentation 53
multi-objective evolutionary algorithms 388
multi-objective optimization 357, 364
multi-objective problems 367
MUSIG activation function 53, 54, 55, 56, 57, 59, 60, 61

nearest neighbor 213, 215, 217
neural networks 93, 116, 486, 487, 488
neuro-fuzzy classifiers 306
nitric acid plant modeling 63
noise 499, 501, 502, 506
NOx reduction 63
numerical function optimization 389

object recognition 499, 500, 506
oil spill management 3
optimization 379, 380, 381, 384, 387, 308
optimization method 389

p-center problem 348, 349
parallel genetic algorithms 330, 332
pareto frontier 359, 362, 363
pareto-optimality 194, 198
particle swarm optimization 171, 176, 193, 194, 411, 412, 416, 499, 500, 502
PCA 453, 454, 455, 456, 461
peer-to-peer network topology 337
pitch contour 519, 520, 521, 523, 526
polymer 357, 358, 359, 364
possibility measure 285, 286, 288, 289
possibility theory 315
probability collectives 441, 443, 449
project scheduling 316, 323
pruning and classification 223

quality of service 368
quantum evolutionary algorithms 389

radial basis function network 519, 520
real estate management 31, 32, 39
reconfigurable hardware 215
regression analysis 520
Reliability 127, 128, 131, 133, 136
reliability 412, 414, 419, 421
robust optimization 285
rosenbrock function 441, 447, 448, 449
rough sets 23, 24, 25, 26, 27, 28

SAPSO5 423
satellite communication 236, 239
search space transformation 139
sensor fusion 13, 15, 17, 18, 19, 21
sensor networks 347, 348
sensor placement 159, 161, 167
shape from focus 453
SIFT 13, 14, 15, 16, 21
simulated annealing 423
social decision 174
social network 171, 172, 173
socio-economic impact
sonar 13, 14, 15, 16, 19
spread spectrum audio watermarking 463, 470
structural control 160, 163, 167
structural design 160
sub-gaussian sources 491
super-gaussian sources 491, 494
SUPER-SAPSO 423, 424, 425, 426, 429, 430
support vector machines 240
surface computation 139
SVD 83, 84, 87, 88, 91
SVM 431, 433, 435, 439, 519, 521
switched linear system
switching signal 399, 400, 401, 402, 404, 407

TCP networks 93, 94, 100, 115, 116, 122
text categorization 223, 224, 225, 230
text mining 223
time series 83, 84, 91

uncertainty 259
user preferences 379, 380, 384, 388

variable precision logic 327, 328, 334
verbal decision analisys 243
virtual community 171, 172, 173, 174, 179
voronoi diagram 348, 349, 353, 354, 355

watermarking 463, 464, 465, 466, 468, 470
wave impact tester 85
wavelet neural network 73, 74
web intelligence 31
web2.0 171, 172

x-ray structure elucidation 181