Leszek Rutkowski Marcin Korytkowski
Rafał Scherer Ryszard Tadeusiewicz
Lotfi A. Zadeh Jacek M. Zurada (Eds.)

LNAI 7895

Artificial Intelligence and Soft Computing

12th International Conference, ICAISC 2013
Zakopane, Poland, June 2013
Proceedings, Part II

Springer

Lecture Notes in Artificial Intelligence 7895

Subseries of Lecture Notes in Computer Science

Leszek Rutkowski Marcin Korytkowski
Rafał Scherer Ryszard Tadeusiewicz
Lotfi A. Zadeh Jacek M. Zurada (Eds.)

Artificial Intelligence and Soft Computing

12th International Conference, ICAISC 2013
Zakopane, Poland, June 9-13, 2013
Proceedings, Part II

Springer

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Leszek Rutkowski
Marcin Korytkowski
Rafał Scherer
Częstochowa University of Technology, Poland
E-mail: {leszek.rutkowski, marcin.korytkowski, rafal.scherer}@iisi.pcz.pl

Ryszard Tadeusiewicz
AGH University of Science and Technology, Kraków, Poland
E-mail: rtad@agh.edu.pl

Lotfi A. Zadeh
University of California, Berkeley, CA, USA
E-mail: zadeh@cs.berkeley.edu

Jacek M. Zurada
University of Louisville, KY, USA
E-mail: jacek.zurada@louisville.edu

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-38609-1 e-ISBN 978-3-642-38610-7
DOI 10.1007/978-3-642-38610-7
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013938841

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

LNCS Sublibrary: SL 7 – Artificial Intelligence

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

Printed on acid-free paper

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

Preface

This volume constitutes the proceedings of the 12th International Conference on Artificial Intelligence and Soft Computing, ICAISC 2013, held in Zakopane, Poland, during June 9–13, 2013. The conference was organized by the Polish Neural Network Society in cooperation with the Academy of Management in Łódź (SWSPiZ), the Department of Computer Engineering at the Czestochowa University of Technology, and the IEEE Computational Intelligence Society, Poland Chapter. Previous conferences took place in Kule (1994), Szczyrk (1996), Kule (1997), and Zakopane (1999, 2000, 2002, 2004, 2006, 2008, 2010, and 2012) and attracted a large number of papers and internationally recognized speakers: Lotfi A. Zadeh, Igor Aizenberg, Shun-ichi Amari, Daniel Amit, Piero P. Bonissone, Jim Bezdek, Zdzislaw Bubnicki, Andrzej Cichocki, Wlodzislaw Duch, Pablo A. Estévez, Jerzy Grzymala-Busse, Martin Hagan, Akira Hirose, Kaoru Hirota, Janusz Kacprzyk, Jim Keller, Laszlo T. Koczy, Soo-Young Lee, Robert Marks, Prof. Evangelia Micheli-Tzanakou, Kaisa Miettinen, Ngoc Thanh Nguyen, Erkki Oja, Witold Pedrycz, Marios M. Polycarpou, José C. Príncipe, Jagath C. Rajapakse, Sarunas Raudys, Enrique Ruspini, Jorg Siekman, Roman Slowinski, Igor Spiridonov, Ryszard Tadeusiewicz, Shiro Usui, Jun Wang, Ronald Y. Yager, Syozo Yasui, and Jacek Zurada. The aim of this conference is to build a bridge between traditional artificial intelligence techniques and recently developed soft computing techniques. It was pointed out by Lotfi A. Zadeh that: "Soft computing (SC) is a coalition of methodologies which are oriented toward the conception and design of information/intelligent systems. The principal members of the coalition are: fuzzy logic (FL), neurocomputing (NC), evolutionary computing (EC), probabilistic computing (PC), chaotic computing (CC), and machine learning (ML). The constituent methodologies of SC are, for the most part, complementary and synergistic rather than competitive." This volume presents both traditional artificial intelligence methods and soft computing techniques. This volume is divided into six parts:

- Evolutionary Algorithms and Their Applications
- Data Mining
- Bioinformatics and Medical Applications
- Agent Systems, Robotics and Control
- Artificial Intelligence in Modeling and Simulation
- Various Problems of Artificial Intelligence

The conference attracted 274 submissions from 27 countries and after the review process, 112 papers were accepted for publication. I would like to thank our participants, invited speakers, and reviewers of the papers for their scientific and personal contribution to the conference. The reviewers listed herein were very helpful in reviewing the papers.

Finally, I thank my co-workers Łukasz Bartczuk, Piotr Dziwiński, Marcin Gabryel, Marcin Korytkowski, and the conference secretary Rafał Scherer, for their enormous efforts to make the conference a very successful event. Moreover, I would like to acknowledge the work of Marcin Korytkowski, who designed the Internet submission system.

June 2013 Leszek Rutkowski

Organization

ICAISC 2013 was organized by the Polish Neural Network Society in cooperation with the SWSPiZ Academy of Management in Łódź, the Department of Computer Engineering at Częstochowa University of Technology, and the IEEE Computational Intelligence Society, Poland Chapter and with technical sponsorship of the IEEE Computational Intelligence Society.

ICAISC Chairs

Honorary chairs	Lotfi Zadeh (USA)
	Jacek Żurada (USA)
General chair	Leszek Rutkowski (Poland)
Co-chairs	Włodzisław Duch (Poland)
	Janusz Kacprzyk (Poland)
	Józef Korbicz (Poland)
	Ryszard Tadeusiewicz (Poland)

ICAISC Program Committee

Rafał Adamczak, Poland
Cesare Alippi, Italy
Shun-ichi Amari, Japan
Rafal A. Angryk, USA
Jarosław Arabas, Poland
Robert Babuska, The Netherlands
Ildar Z. Batyrshin, Russia
James C. Bezdek, USA
Marco Block-Berlitz, Germany
Leon Bobrowski, Poland
Leonard Bolc, Poland
Piero P. Bonissone, USA
Bernadette Bouchon-Meunier, France
James Buckley, Poland
Tadeusz Burczynski, Poland
Andrzej Cader, Poland
Juan Luis Castro, Spain
Yen-Wei Chen, Japan
Wojciech Cholewa, Poland
Fahmida N. Chowdhury, USA
Andrzej Cichocki, Japan
Paweł Cichosz, Poland
Krzysztof Cios, USA
Ian Cloete, Germany
Oscar Cordón, Spain
Bernard De Baets, Belgium
Nabil Derbel, Tunisia
Ewa Dudek-Dyduch, Poland
Ludmiła Dymowa, Poland
Andrzej Dzieliński, Poland
David Elizondo, UK
Meng Joo Er, Singapore
Pablo Estevez, Chile
János Fodor, Hungary
David B. Fogel, USA
Roman Galar, Poland
Alexander I. Galushkin, Russia
Adam Gaweda, USA
Joydeep Ghosh, USA
Juan Jose Gonzalez de la Rosa, Spain
Marian Bolesław Gorzałczany, Poland
Krzysztof Grąbczewski , Poland
Garrison Greenwood, USA
Jerzy W. Grzymala-Busse, USA

Hani Hagras, UK
Saman Halgamuge, Australia
Rainer Hampel, Germany
Zygmunt Hasiewicz, Poland
Yoichi Hayashi, Japan
Tim Hendtlass, Australia
Francisco Herrera, Spain
Kaoru Hirota, Japan
Adrian Horzyk, Poland
Tingwen Huang, USA
Hisao Ishibuchi, Japan
Mo Jamshidi, USA
Andrzej Janczak, Poland
Norbert Jankowski, Poland
Robert John, UK
Jerzy Józefczyk, Poland
Tadeusz Kaczorek, Poland
Władysław Kamiński, Poland
Nikola Kasabov, New Zealand
Okyay Kaynak, Turkey
Vojislav Kecman, New Zealand
James M. Keller, USA
Etienne Kerre, Belgium
Frank Klawonn, Germany
Jacek Kluska, Poland
Leonid Kompanets, Poland
Przemysław Korohoda, Poland
Jacek Koronacki, Poland
Witold Kosiński, Poland
Jan M. Kościelny, Poland
Zdzisław Kowalczuk, Poland
Robert Kozma, USA
László Kóczy, Hungary
Rudolf Kruse, Germany
Boris V. Kryzhanovsky, Russia
Adam Krzyzak, Canada
Juliusz Kulikowski, Poland
Roman Kulikowski, Poland
Věra Kůrková, Czech Republic
Marek Kurzyński, Poland
Halina Kwaśnicka, Poland
Soo-Young Lee, Korea
George Lendaris, USA
Antoni Ligęza, Poland
Sławomir Litwiński, Poland
Zhi-Qiang Liu, Hong Kong
Simon M. Lucas, UK
Jacek Łęski, Poland
Bohdan Macukow, Poland
Kurosh Madani, France
Luis Magdalena, Spain
Witold Malina , Poland
Krzysztof Malinowski, Poland
Jacek Mańdziuk, Poland
Antonino Marvuglia, Ireland
Andrzej Materka, Poland
Jaroslaw Meller, Poland
Jerry M. Mendel, USA
Radko Mesiar, Slovakia
Zbigniew Michalewicz, Australia
Zbigniew Mikrut, Poland
Sudip Misra, USA
Wojciech Moczulski, Poland
Javier Montero, Spain
Eduard Montseny, Spain
Kazumi Nakamatsu, Japan
Detlef D. Nauck, Germany
Antoine Naud, Poland
Edward Nawarecki, Poland
Ngoc Thanh Nguyen, Poland
Antoni Niederliński, Poland
Robert Nowicki, Poland
Andrzej Obuchowicz, Poland
Marek Ogiela, Poland
Erkki Oja, Finland
Stanisław Osowski, Poland
Nikhil R. Pal, India
Maciej Patan, Poland
Witold Pedrycz, Canada
Leonid Perlovsky, USA
Andrzej Pieczyński, Poland
Andrzej Piegat, Poland
Vincenzo Piuri, Italy
Lech Polkowski, Poland
Marios M. Polycarpou, Cyprus
Danil Prokhorov, USA
Anna Radzikowska, Poland
Ewaryst Rafajłowicz, Poland
Sarunas Raudys, Lithuania
Olga Rebrova, Russia

Vladimir Red'ko, Russia
Raúl Rojas, Germany
Imre J. Rudas, Hungary
Enrique H. Ruspini, USA
Khalid Saeed, Poland
Dominik Sankowski, Poland
Norihide Sano, Japan
Robert Schaefer, Poland
Rudy Setiono, Singapore
Paweł Sewastianow, Poland
Jennie Si , USA
Peter Sincak, Slovakia
Andrzej Skowron, Poland
Ewa Skubalska-Rafajłowicz, Poland
Roman Słowiński, Poland
Tomasz G. Smolinski, USA
Czesław Smutnicki, Poland
Pilar Sobrevilla, Spain
Janusz Starzyk, USA
Jerzy Stefanowski, Poland
Pawel Strumillo, Poland
Ron Sun, USA
Johan Suykens Suykens, Belgium
Piotr Szczepaniak, Poland
Eulalia J. Szmidt, Poland
Przemysław Śliwiński, Poland
Adam Słowik, Poland
Jerzy Świątek , Poland
Hideyuki Takagi, Japan
Yury Tiumentsev, Russia
Vicenç Torra, Spain
Burhan Turksen, Canada
Shiro Usui, Japan
Michael Wagenknecht, Germany
Tomasz Walkowiak, Poland
Deliang Wang, USA
Jun Wang, Hong Kong
Lipo Wang, Singapore
Zenon Waszczyszyn, Poland
Paul Werbos, USA
Slawo Wesolkowski, Canada
Sławomir Wiak, Poland
Bernard Widrow, USA
Kay C. Wiese, Canada
Bogdan M. Wilamowski, USA
Donald C. Wunsch, USA
Maciej Wygralak, Poland
Roman Wyrzykowski, Poland
Ronald R. Yager, USA
Xin-She Yang, United Kingdom
Gary Yen, USA
John Yen, USA
Sławomir Zadrożny, Poland
Ali M. S. Zalzala, United Arab Emirates

ICAISC Organizing Committee

Rafał Scherer, Secretary
Łukasz Bartczuk, Organizing Committee Member
Piotr Dziwiński, Organizing Committee Member
Marcin Gabryel, Finance Chair
Marcin Korytkowski, Databases and Internet Submissions

Additional Reviewers

R. Adamczak
M. Al-Dhelaan
S. Amari
A. Bari
Ł. Bartczuk
A. Bielskis
M. Blachnik
L. Bobrowski
P. Boguś
A. Borkowski
B. Bouchon-Meunier
W. Bozejko
T. Burczyński
R. Burduk
B. Butkiewicz

K. Cetnarowicz
M. Chang
L. Chmielewski
M. Choraś
R. Cierniak
B. Cyganek
I. Czarnowski
J. de la Rosa
K. Dembczynski
L. Diosan
E. Dudek-Dyduch
L. Dymowa
A. Dzieliński
P. Dziwiński
S. Ehteram
D. Elizondo
A. Fanea
M. Flasiński
M. Gabryel
S. Gadepally
F. Gomide
M. Gorzałczany
E. Grabska
K. Grąbczewski
W. Greblicki
J. Grzymala-Busse
R. Hampel
Z. Hasiewicz
Y. Hayashi
T. Hendtlass
Z. Hendzel
K. Hirota
A. Horzyk
E. Hrynkiewicz
D. Jakóbczak
A. Janczak
D. Kacprzak
T. Kaczorek
W. Kamiński
V. Kecman
E. Kerre
P. Klęsk
J. Kluska
L. Koczy
Z. Kokosinski
A. Kołakowska
J. Korbicz
P. Korohoda
J. Koronacki
M. Korytkowski
M. Korzeń
W. Kosiński
J. Kościelny
L. Kotulski
J. Kozlak
M. Kretowski
R. Kruse
J. Kulikowski
V. Kurkova
M. Kurzyńñski
J. Kusiak
S. Lee
A. Ligęza
S. Litwiński
I. Lovtsova
J. Łęski
B. Macukow
K. Madani
W. Malina
J. Mańdziuk
U. Markowska-Kaczmar
A. Marszałek
A. Martin
A. Materka
R. Matuk Herrera
J. Mendel
J. Michalkiewicz
W. Mitkowski
M. Morzy
M. Nieniewski
R. Nowicki
A. Obuchowicz
E. Oja
S. Osowski
K. Patan
S. Paul
A. Pieczyński
A. Piegat
V. Piuri
P. Prokopowicz
A. Przybył
E. Rafajłowicz
E. Rakus-Andersson
F. Rastegar
L. Rolka
I. Rudas
F. Rudziński
A. Rusiecki
L. Rutkowski
H. Safari
N. Sano
R. Scherer
P. Sevastjanov
A. Sędziwy
L. Singh
W. Skarbek
A. Skowron
E. Skubalska-Rafajłowicz
K. Slot
A. Słowik
C. Smutnicki
A. Sokołowski
T. Sołtysiński
B. Starosta
J. Stefanowski
B. Strug
P. Strumiłło
P. Sudhish
J. Swacha
E. Szmidt
J. Świątek
R. Tadeusiewicz
H. Takagi
Y. Tiumentsev
V. Torra
B. Trawinski
M. Wagenknecht
T. Walkowiak
H. Wang
J. Wąs
M. Witczak
M. Wozniak
M. Wygralak
R. Wyrzykowski
J. Zabrodzki

Table of Contents – Part II

I Evolutionary Algorithms and Their Applications

Global Induction of Oblique Model Trees: An Evolutionary Approach . . . 1
Marcin Czajkowski and Marek Kretowski

Genetic Cost Optimization of the *GI/M/1/N* Finite-Buffer Queue with a Single Vacation Policy . . . 12
Marcin Gabryel, Robert K. Nowicki, Marcin Woźniak, and Wojciech M. Kempa

Hybrid Genetic-Fuzzy Algorithm for Variable Selection in Spectroscopy . . . 24
Telma Woerle de Lima, Anderson da Silva Soares, Clarimar José Coelho, Rogério Lopes Salvini, and Gustavo Teodoro Laureano

Multiple Choice Strategy – A Novel Approach for Particle Swarm Optimization – Preliminary Study . . . 36
Michal Pluhacek, Roman Senkerik, and Ivan Zelinka

Method of Handling Constraints in Differential Evolution Using Fletcher's Filter . . . 46
Wojciech Rafajłowicz

Chaos Driven Differential Evolution with Lozi Map in the Task of Chemical Reactor Optimization . . . 56
Roman Senkerik, Donald Davendra, Ivan Zelinka, Michal Pluhacek, and Zuzana Kominkova Oplatkova

Application of Geometric Differential Evolution Algorithm to Design Minimal Phase Digital Filters with Atypical Characteristics for Their Hardware or Software Implementation . . . 67
Adam Slowik

Bio-inspired Optimization of Thermomechanical Structures . . . 79
Mirosław Szczepanik, Arkadiusz Poteralski, Adam Długosz, Wacław Kuś, and Tadeusz Burczyński

Some Aspects of Evolutionary Designing Optimal Controllers . . . 91
Jacek Szczypta, Andrzej Przybył, and Krzysztof Cpałka

The Niching Mechanism in the Evolutionary Method of Path Planning 101
Roman Śmierzchalski, Piotr Kolendo, Lukasz Kuczkowski, Bartosz Jaworski, and Anna Witkowska

New Algorithm for Evolutionary Selection of the Dynamic Signature Global Features 113
Marcin Zalasiński, Krystian Łapa, and Krzysztof Cpałka

On the Performance of Master-Slave Parallelization Methods for Multi-Objective Evolutionary Algorithms 122
Alexandru-Ciprian Zăvoianu, Edwin Lughofer, Werner Koppelstätter, Günther Weidenholzer, Wolfgang Amrhein, and Erich Peter Klement

II Data Mining

TRCM: A Methodology for Temporal Analysis of Evolving Concepts in Twitter 135
Mariam Adedoyin-Olowe, Mohamed Medhat Gaber, and Frederic Stahl

Identifying Uncertain Galaxy Morphologies Using Unsupervised Learning 146
Kieran Jay Edwards and Mohamed Medhat Gaber

Kernel Estimation of Regression Functions in the Boundary Regions 158
Tomasz Gałkowski

Instance Selection in Logical Rule Extraction for Regression Problems 167
Mirosław Kordos, Szymon Białka, and Marcin Blachnik

On Perturbation Measure of Clusters: Application 176
Maciej Krawczak and Grażyna Szkatuła

Using Topology Preservation Measures for Multidimensional Intelligent Data Analysis in the Reduced Feature Space 184
Szymon Łukasik and Piotr Kulczycki

HRoBi – The Algorithm for Hierarchical Rough Biclustering 194
Marcin Michalak and Magdalena Stawarz

Proximity Measures and Results Validation in Biclustering – A Survey 206
Patryk Orzechowski

Clustering and Classification of Time Series Representing Sign Language Words 218
Mariusz Oszust and Marian Wysocki

Evaluation of the Mini-Models Robustness to Data Uncertainty with the Application of the Information-Gap Theory 230
Marcin Pluciński

A Clustering Method Based on the Modified RS Validity Index 242
Artur Starczewski

III Bioinformatics, Biometrics and Medical Applications

Online Emotion Classification from Electroencephalographic Signals : A First Study Conducted in a Realistic Movie Theater 251
Thierry Castermans, Matthieu Duvinage, and Nicolas Riche

White Blood Cell Differential Counts Using Convolutional Neural Networks for Low Resolution Images 263
Mehdi Habibzadeh, Adam Krzyżak, and Thomas Fevens

Users Verification Based on Palm-Prints and Hand Geometry with Hidden Markov Models 275
Mariusz Kubanek, Dorota Smorawa, and Lukasz Adrjanowicz

A New Approach to Determine Three-Dimensional Facial Landmarks . . . 286
Sebastian Pabiasz and Janusz T. Starczewski

A Comparison of Dimensionality Reduction Techniques in Virtual Screening 297
Kitsuchart Pasupa

Improved X-ray Edge Detection Based on Background Extraction Algorithm 309
Jakub Romanowski, Tomasz Nowak, Patryk Najgebauer, and Sławomir Litwiński

Comparison of Time-Frequency Feature Extraction Methods for EEG Signals Classification 320
Grzegorz Rutkowski, Krzysztof Patan, and Paweł Leśniak

GP-Pi: Using Genetic Programming with Penalization and Initialization on Genome-Wide Association Study 330
Ho-Yin Sze-To, Kwan-Yeung Lee, Kai-Yuen Tso, Man-Hon Wong, Kin-Hong Lee, Nelson L.S. Tang, and Kwong-Sak Leung

New Approach for the On-Line Signature Verification Based on Method of Horizontal Partitioning 342
Marcin Zalasiński and Krzysztof Cpałka

IV Agent Systems, Robotics and Control

Adaptive Critic Designs in Control of Robots Formation in Unknown Environment 351
Zenon Hendzel, Andrzej Burghardt, and Marcin Szuster

State-Space Reduction through Preference Modeling 363
Radosław Klimek, Igor Wojnicki, and Sebastian Ernst

Multi-Agent Temporary Logic $TS4^{U}_{K_n}$ Based at Non-linear Time and Imitating Uncertainty via Agents' Interaction 375
David McLean and Vladimir Rybakov

Opponent Modelling by Sequence Prediction and Lookahead in Two-Player Games 385
Richard Mealing and Jonathan L. Shapiro

Supporting Fault Tolerance in Graph-Based Multi-agent Computations 397
Adam Sędziwy and Leszek Kotulski

moviQuest-MAS: An Intelligent Platform for Ubiquitous Social Networking Business 407
Ramon Soto, Alexandro Soto, and Juan Camalich

V Artificial Intelligence in Modeling and Simulation

Substitution Tasks Method for Discrete Optimization 419
Ewa Dudek-Dyduch and Lidia Dutkiewicz

Inverse Continuous Casting Problem Solved by Applying the Artificial Bee Colony Algorithm 431
Edyta Hetmaniok, Damian Słota, Adam Zielonka, and Mariusz Pleszczyński

Stock Trading with Random Forests, Trend Detection Tests and Force Index Volume Indicators 441
Piotr Ładyżyński, Kamil Żbikowski, and Przemysław Grzegorzewski

Proposal of an Inference Engine Architecture for Business Rules and Processes 453
Grzegorz J. Nalepa, Krzysztof Kluza, and Krzysztof Kaczor

Emergence of New Global Airline Networks and Distributing Loads in Star Airports 465
Hidefumi Sawai

VI Various Problems of Artificial Intelligence

Selection of Relevant Features for Text Classification with K-NN 477
Jerzy Balicki, Henryk Krawczyk, Łukasz Rymko, and Julian Szymański

Parallel Neuro-Tabu Search Algorithm for the Job Shop Scheduling Problem 489
Wojciech Bożejko, Mariusz Uchroński, and Mieczysław Wodecki

Matrix Factorization for Travel Time Estimation in Large Traffic Networks 500
Krzysztof Dembczyński, Wojciech Kotłowski, Przemysław Gaweł, Adam Szarecki, and Andrzej Jaszkiewicz

Semantically-Driven Rule Interoperability – Concept Proposal 511
Krzysztof Kaczor and Grzegorz J. Nalepa

A New Approach to Designing Interpretable Models of Dynamic Systems 523
Krystian Łapa, Andrzej Przybył, and Krzysztof Cpałka

Towards Efficient Information Exchange in Fusion Networks 535
Hela Masri, Adel Guitouni, and Saoussen Krichen

Application of Particle Swarm Optimization Algorithm to Dynamic Vehicle Routing Problem 547
Michał Okulewicz and Jacek Mańdziuk

Distribution Network Characteristics in Rough Set Approach 559
Henryk Piech and Aleksandra Ptak

Comparison between PSO and AIS on the Basis of Identification of Material Constants in Piezoelectrics 569
Arkadiusz Poteralski, Mirosław Szczepanik, Grzegorz Dziatkiewicz, Wacław Kuś, and Tadeusz Burczyński

Universal Intelligence, Creativity, and Trust in Emerging Global Expert Systems 582
Andrzej M.J. Skulimowski

A Swarm Intelligence Approach to Flexible Job-Shop Scheduling Problem with No-Wait Constraint in Remanufacturing 593
Shyam Sundar, P.N. Suganthan, and T.J. Chua

Uniform Approach to Concept Interpretation, Active Contour Methods and Case-Based Reasoning 603
Piotr S. Szczepaniak and Arkadiusz Tomczyk

Modified Merge Sort Algorithm for Large Scale Data Sets 612
Marcin Woźniak, Zbigniew Marszałek, Marcin Gabryel, and Robert K. Nowicki

Table of Contents – Part I

I Neural Networks and Their Applications

Neural Data Analysis: Ensemble Neural Network Rule Extraction Approach and Its Theoretical and Historical Backgrounds (Invited paper) 1
Yoichi Hayashi

A New Method of Centers Location in Gaussian RBF Interpolation Networks 20
Marek Bazan and Ewa Skubalska-Rafajłowicz

Parallel Approach to Learning of the Recurrent Jordan Neural Network 32
Jarosław Bilski and Jacek Smoląg

Agent-Based Population Learning Algorithm for RBF Network Tuning 41
Ireneusz Czarnowski and Piotr Jędrzejowicz

Forecasting Time Series with Multiple Seasonal Cycles Using Neural Networks with Local Learning 52
Grzegorz Dudek

Reinforcement Learning in Discrete Neural Control of the Underactuated System 64
Zenon Hendzel, Andrzej Burghardt, and Marcin Szuster

Associative Text Representation and Correction 76
Adrian Horzyk and Marcin Gadamer

Some Aspects of Neural Network State Variable Estimator Improvement in Induction Motor Drive 88
Jerzy Jelonkiewicz and Łukasz Laskowski

The Echo State Network on the Graphics Processing Unit 96
Tūreiti Keith and Stephen J. Weddell

Solution of Inverse Kinematics by PSO Based on Split and Merge of Particles 108
Koji Kinoshita, Kenji Murakami, and Masaharu Isshiki

Probabilistic Neural Network Structure Reduction for Medical Data Classification 118
Maciej Kusy and Jacek Kluska

Development of Explicit Neural Predictive Control Algorithm Using Particle Swarm Optimisation 130
Maciej Ławryńczuk

Designing of State-Space Neural Model and Its Application to Robust Fault Detection 140
Marcin Mrugalski

Employing Self-Organizing Map for Fraud Detection 150
Dominik Olszewski, Janusz Kacprzyk, and Sławomir Zadrożny

A Study on the Scalability of Artificial Neural Networks Training Algorithms Using Multiple-Criteria Decision-Making Methods 162
Diego Peteiro-Barral and Bertha Guijarro-Berdiñas

Testing the Generalization of Feedforward Neural Networks with Median Neuron Input Function 174
Andrzej Rusiecki

Biological Plausibility in an Artificial Neural Network Applied to Real Predictive Tasks 183
Alberione Braz da Silva and João Luís Garcia Rosa

Random Sieve Based on Projections for RBF Neural Net Structure Selection 193
Ewa Skubalska-Rafajłowicz and Ewaryst Rafajłowicz

Eigenvalue Spectra of Functional Networks in fMRI Data and Artificial Models 205
Katarzyna Zając and Jarosław Piersa

II Fuzzy Systems and Their Applications

A Multi-objective Subtractive FCM Based TSK Fuzzy System with Input Selection, and Its Application to Dynamic Inverse Modelling of MR Dampers 215
Mohsen Askari, Jianchan Li, and Bijan Samali

Hybrid State Variables - Fuzzy Logic Modelling of Nonlinear Objects . . . 227
Łukasz Bartczuk, Andrzej Przybył, and Piotr Dziwiński

Properties of Plausibility Conflict of Belief Functions 235
Milan Daniel

The Use of Intuitionistic Fuzzy Values in Rule-Base Evidential Reasoning 247
Ludmila Dymova, Pavel Sevastjanov, and Kamil Tkacz

Financial Stock Data and Ordered Fuzzy Numbers 259
Dariusz Kacprzak, Witold Kosiński, and W. Konrad Kosiński

Diversity of Opinion Evaluated by Ordered Fuzzy Numbers 271
Magdalena Kacprzak, Witold Kosiński, and Katarzyna Węgrzyn-Wolska

Hardware Implementation of P1-TS Fuzzy Rule-Based Systems on FPGA 282
Jacek Kluska and Zbigniew Hajduk

Evolutionary Strategy for the Fuzzy Flip-Flop Neural Networks Supervised Learning Procedure 294
Piotr A. Kowalski

Air Traffic Incidents Analysis with the Use of Fuzzy Sets 306
Michał Lower, Jan Magott, and Jacek Skorupski

Automatic Design of Interpretable Fuzzy Partitions with Variable Granularity: An Experimental Comparison 318
Marco Lucarelli, Ciro Castiello, Anna Maria Fanelli, and Corrado Mencar

A New Method for Designing and Complexity Reduction of Neuro-fuzzy Systems for Nonlinear Modelling 329
Krystian Łapa, Marcin Zalasiński, and Krzysztof Cpałka

Modelling Financial High Frequency Data Using Ordered Fuzzy Numbers 345
Adam Marszałek and Tadeusz Burczyński

The Modified Sequential-Binary Approach for Fuzzy Operations on Correlated Assessments 353
Jacek Pietraszek

Flexible and Simple Methods of Calculations on Fuzzy Numbers with the Ordered Fuzzy Numbers Model 365
Piotr Prokopowicz

The Use of Fuzzy Numbers in the Process of Designing Relational Fuzzy Cognitive Maps 376
Grzegorz Słoń

Metasets, Intuitionistic Fuzzy Sets and Uncertainty 388
Bartłomiej Starosta and Witold Kosiński

A New Similarity Function for Generalized Trapezoidal Fuzzy Numbers 400
E. Vicente, A. Mateos, and A. Jiménez

Fuzzy Granulation Approach to Color Digital Picture Recognition 412
Krzysztof Wiaderek and Danuta Rutkowska

III Pattern Classification

Pruning One-Class Classifier Ensembles by Combining Sphere Intersection and Consistency Measures 426
Bartosz Krawczyk and Michał Woźniak

SUCCESS: A New Approach for Semi-supervised Classification of Time-Series 437
Kristóf Marussy and Krisztian Buza

A New Method of Improving Classification Accuracy of Decision Tree in Case of Incomplete Samples 448
Bartosz A. Nowak, Robert K. Nowicki, and Wojciech K. Mleczko

Adaptation of Decision Trees for Handling Concept Drift 459
Lena Pietruczuk, Piotr Duda, and Maciej Jaworski

Hesitant Neural Gas for Supervised and Semi-supervised Classification 474
Piotr Płoński and Krzysztof Zaremba

Intuitionistic Fuzzy Classifier for Imbalanced Classes 483
Eulalia Szmidt, Janusz Kacprzyk, and Marta Kukier

Novel Algorithm for the On-Line Signature Verification Using Selected Discretization Points Groups 493
Marcin Zalasiński and Krzysztof Cpałka

IV Computer Vision, Image and Speech Analysis

Speech Enhancement in Noisy Environments in Hearing Aids Driven by a Tailored Gain Function Based on a Gaussian Mixture Model 503
Lorena Álvarez, Enrique Alexandre, Cosme Llerena, Roberto Gil-Pita, and Lucas Cuadra

Two-Dimensional Hidden Markov Models for Pattern Recognition 515
Janusz Bobulski and Lukasz Adrjanowicz

Video Compression Algorithm Based on Neural Networks 524
Robert Cierniak and Michal Knop

Online Crowdsource System Supporting Ground Truth Datasets Creation 532
Paweł Drozda, Krzysztof Sopyła, and Przemysław Górecki

Object Detection by Simple Fuzzy Classifiers Generated by Boosting ... 540
Marcin Gabryel, Marcin Korytkowski, Rafał Scherer, and Leszek Rutkowski

Road Surface Recognition System Based on Its Picture 548
Michal Jankowski and Jacek Mazurkiewicz

Landmine Detection in 3D Images from Ground Penetrating Radar Using Haar-Like Features 559
Przemysław Klęsk, Andrzej Godziuk, Mariusz Kapruziak, and Bogdan Olech

Synchronizing Speech Mixtures in Speech Separation Problems under Reverberant Conditions 568
Cosme Llerena, Roberto Gil-Pita, Lorena Álvarez, and Manuel Rosa-Zurera

Neuronal Model-Based Image Reconstruction from Projections Method 580
Anna Lorent, Michał Knaś, and Piotr Dobosz

Representation of Edge Detection Results Based on Graph Theory 588
Patryk Najgebauer, Tomasz Nowak, Jakub Romanowski, Janusz Rygał, and Marcin Korytkowski

A Novel Graph-Based Descriptor for Object Matching 602
Tomasz Nowak, Patryk Najgebauer, Janusz Rygał, and Rafał Scherer

Extraction of Objects from Images Using Density of Edges as Basis for GrabCut Algorithm 613
Janusz Rygał, Patryk Najgebauer, Jakub Romanowski, and Rafał Scherer

Computer Vision Based Method for Real Time Material and Structure Parameters Estimation Using Digital Image Correlation, Particle Filtering and Finite Element Method 624
Marcin Tekieli and Marek Słoński

Author Index 635

Global Induction of Oblique Model Trees: An Evolutionary Approach

Marcin Czajkowski and Marek Kretowski

Faculty of Computer Science, Bialystok University of Technology
Wiejska 45a, 15-351 Białystok, Poland
{m.czajkowski,m.kretowski}@pb.edu.pl

Abstract. In this paper we propose a new evolutionary algorithm for global induction of oblique model trees that associates leaves with multiple linear regression models. In contrast to the typical top-down approaches it globally searches for the best tree structure, splitting hyperplanes in internal nodes and models in the leaves. The general structure of proposed solution follows a typical framework of evolutionary algorithms with an unstructured population and a generational selection. We propose specialized genetic operators to mutate and cross-over individuals (trees). The fitness function is based on the Bayesian Information Criterion. In preliminary experimental evaluation we show the impact of the tree representation on solving different prediction problems.

Keywords: data mining, evolutionary algorithms, model trees, oblique trees, global induction.

1 Introduction

Decision trees [26] are one of the most popular and frequently applied prediction technique in classification and regression problems. Regression and model trees are now popular alternatives to classical statistical techniques like standard regression or logistic regression [12].

In this paper we want to investigate a global approach to oblique model tree induction based on a specialized evolutionary algorithm. This solution extends our previous research on evolutionary univariate regression and model trees.

1.1 Background

Data mining [10] is a process of extracting useful information, relationships and hidden patterns from large databases. The tree-based approaches are gaining in popularity because they are easy to interpret, visualize and their decisions can be easily explained. The ease of application, fast operation and what may be the most important, the effectiveness of decision trees, makes them powerful and popular tool [14]. Decision trees are also applicable when the data does not satisfy rigorous assumptions required by more traditional methods [12].

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 1–11, 2013.

The problem of learning the optimal decision tree is known to be NP-complete [21]. Consequently, classical decision-tree learning algorithms are based on heuristics such as greedy top-down approach [25]. Starting from the root node it searches for the locally optimal split (test) according to the given optimality measure. Next, the training data is redirected to newly created nodes. This procedure is recursively repeated until stopping criteria are met. Finally, the post-pruning is applied to improve generalization power of the predictive model.

Most of the tree inducing algorithms partition the feature space with axis-parallel hyperplanes. These types of trees are called univariate decision trees since each split in non-terminal node involves a single feature. For continuous-valued features usually inequality tests with binary outcomes are applied and for nominal features mutually exclusive groups of feature values are associated with the outcomes. When more than one feature is taken into account to build a test in internal node, then we deal with multivariate decision trees. The most common form of such test is an oblique split, which is based on linear combination of features (hyper-plane). The decision tree which applies only oblique tests is often called oblique or linear, whereas heterogeneous trees with univariate, linear and other multivariate (e.g., instance-based) tests can be called mixed decision trees [18]. It should be emphasized that computational complexity of the multivariate induction is generally significantly higher than the univariate induction.

Regression and model trees [12] may be considered as a variant of decision trees, designed to approximate real-valued functions instead of being used for classification tasks. The main difference between regression tree and model tree is that, in the latter, constant value in the terminal node is replaced by a regression plane. Each leaf of the model tree holds a linear (or nonlinear) model whose output is the final prediction value. One of the first and most known regression tree solutions is the *CART* system [4]. It finds a split that minimizes the sum of squared residuals and builds a piecewise constant model with each terminal node fitted by the training sample mean. The accuracy of prediction was later improved by replacing single values in the leaves by more advanced models: *M5* [29] proposed by Quinlan, induces a univariate model tree that contains at leaves multiple linear models analogous to piecewise linear functions; *HTL* presented in [28] goes further and evaluates linear and nonlinear models in the terminal nodes.

1.2 Motivation

The linear regression is a global model in which a single predictive function holds over the entire data-space [12]. However, many regression problems cannot be solved by single regression models especially when the data has many features which interact in complicated ways. Recursively partitioning the data and fitting local models to the smaller regions, where the interactions are more simple, is a good alternative to complicated, nonlinear regression approaches. Such technique is fast and generally efficient in many practical problems, but obviously does not guarantee the optimal solution. Due to the greedy nature, algorithms may not generate the smallest possible number of rules for a given problem [22]

and large number of rules results in decreased comprehensibility and often prediction accuracy. Hence, application of evolutionary algorithms (EAs) [20] to the problem of decision tree induction [2] become increasingly popular alternative because instead of local search, EAs can perform a global search in the space of candidate solutions.

In addition, simple univariate tests are convenient, however they generated trees may be complicated and inaccurate if the data is more suitably partitioned by not axel-parallel hyper-planes. Therefore in some domains, oblique trees may result in much smaller and more accurate trees.

In our previous works, we applied EAs to univariate regression [16] and model trees [6] and investigated the impact of memetic extensions [7]. In this paper we extend the GMT solution and search also for splitting hyper-planes in internal nodes. We propose a global approach called oblique Global Model Tree ($oGMT$) which finds accurate and less complex solutions to the popular, greedy counterparts. New specialized operators for the oblique split search are designed and a fitness function that penalizes the tree for over-parametrization and reflects not only the number of nodes but also the complexity of the tests is suitably defined. Previously performed research on oblique classification trees [15] showed that the oblique global algorithm managed to find more compact classifiers than the competitors.

1.3 Related Work

Multiple authors have proposed methods to limit negative effects of inducing the decision tree with greedy strategy. One of the first attempts to optimize the overall model tree error was presented in *RETRIS* [13]. The algorithm simultaneously optimizes the split and the models at the terminal nodes to minimize the global error. However *RETRIS* is not scalable and does not support larger datasets because of its huge complexity [22]. *SMOTI* [19] builds regression models not only in leaves but also in the upper parts of the tree. Authors claim that this allows for individual predictors to have both global and local effects on the model tree. In *LLRT* [30] authors search for optimal solution by a near-exhaustive evaluation of all possible splits in a node, based on the quality of fit of linear regression models in the resulting branches.

In the literature, there are attempts of applying evolutionary approach for induction of decision trees (please refer to survey [2]). In the $TARGET$ solution [9], authors propose to evolve a $CART$-like regression tree with simple operators and the Bayesian Information Criterion (BIC) [27] as a fitness function. Later solutions focus on evolving model trees with linear models: $E-Motion$ [1] and non-linear models: $GPMCC$ [23] in the leaves.

To the best of our knowledge, all evolutionary evolved regression and model trees have univariate tests in the splitting nodes. There are however, few oblique regression trees like: $SECRET$ [8] where authors show that the classification approach which bases on splitting two Gaussian mixtures can find better partitions than the $CART$ system and solution proposed by Li et al. [17] where a linear regression tree algorithm finds oblique splits using Principal Hessian Analysis.

The oblique regression and model trees are not as popular as the axis-parallel because the tests require greater computational resources and are much more difficult to interpret than the axis-parallel algorithms.

2 Global Induction of Oblique Model Trees

In this section we would like to present the *GMT* algorithm for global induction of oblique model trees. Structure of the proposed solution follows a typical framework of evolutionary algorithms [20] with an unstructured population and a generational selection.

2.1 Preliminaries

A learning set is composed of M objects: N-dimensional feature vectors $x_j = [x_{j,1}, \ldots, x_{j,N}]$ $(j = 1, \ldots, M)$, each one with defined dependent variable. The feature space could be divided into two regions by a hyper-plane:

$$H(w, \theta) = \{x : \langle w, x \rangle = \theta\}, \tag{1}$$

where $w = [w_1, \ldots, w_N]$ is a weight vector, θ is a threshold and $\langle w, x \rangle$ represents an inner product. If $\langle w, x_i \rangle - \theta > 0$, it can be said that the feature vector x_i is on the positive side of the hyper-plane $H(w, \theta)$.

A *dipole* [15] is a pair (x_i, x_j) of feature vectors. A *dipole* is called *long* if feature vectors constituting it has much different values of the dependent variable. First feature vector that constitutes dipole is randomly selected from the training objects located in the node. Remaining feature vectors are sorted decreasingly according to the differences between dependent variable values to the selected object. To find a second object that constitutes dipole we applied mechanism similar to the ranking linear selection [20] where the selection of the object depends only on its position in the sorted list and not on the actual value. We situate the feature vectors x_i and x_j on the opposite sides of the dividing hyper-plane. The hyper-plane $H(w, \theta)$ splits the dipole (x_i, x_j) if:

$$(\langle w, x_i \rangle - \theta) * (\langle w, x_j \rangle - \theta) < 0. \tag{2}$$

2.2 Representation

The oblique model trees are represented in their actual form as binary trees with splitting hyper-planes in non-terminal nodes and multiple linear models in the leaves. Each hyper-plane in the tree can be represented by a fixed-size $N + 1$-dimensional table of real numbers corresponding to the vector weight w and threshold θ. Each model in the leaf is constructed using standard regression technique [24] with objects and feature vectors associated with that node. The prediction $y_k(x_j)$ calculated for the k-th leaf and j-th object is explained by a linear combination of multiple independent variables $x_{j,1}, x_{j,2}, \ldots, x_{j,N}$:

$$y_k(x_j) = \upsilon_{k,0} + \upsilon_{k,1} * x_{j,1} + \upsilon_{k,2} * x_{j,2} + \ldots + \upsilon_{k,N} * x_{j,N} \tag{3}$$

where $\upsilon_{k,0}, \ldots, \upsilon_{k,N}$ are fixed coefficients that minimizes the sum of squared residuals of the model in the leaf k.

2.3 Initialization

In general, initial population should be generated randomly to cover the entire range of possible solutions. Due to the large solution space, initial individuals are created by applying the classical top-down algorithm, similarly to the typical approaches like *CART* and *M5*. An effective test in non-terminal node is searched based on randomly chosen *long dipole*. The recursive partitioning can finish:

- on a random depth;
- when all the training objects in node are characterized by the same predicted value (or varies only slightly [29]);
- when the number of objects in a node is lower than the predefined value (default value: 5).

Next, a multiple linear model is built at each terminal node. To prevent the EA from being trapped in local optima, initial individuals are induced on a random subsamples of the training data (10% of data, but not more than 500 objects) and the potential tests in internal nodes are searched on random subsets of features (50% of features).

2.4 Fitness Function

The specification of a suitable fitness function is one of the most important and sensitive element in design of the evolutionary algorithm. The direct minimization of the prediction error measured on the learning set usually leads to the overfitting problem. In this work we continue to use Bayesian information criterion *BIC* [27] as a fitness function [6,7]. The *BIC* formula is given by:

$$Fit_{BIC}(T) = -2 * ln(L(T)) + ln(n) * k(T), \tag{4}$$

where $L(T)$ is maximum of likelihood function of the tree T, $k(T)$ is the number of model parameters and n is the number of observations. The log(likelihood) function $L(T)$ is typical for regression models [11] and can be expressed as:

$$ln(L(T)) = -0.5n * [ln(2\pi) + ln(SS_e(T)/n) + 1], \tag{5}$$

where $SS_e(T)$ is the sum of squared residuals on the training data of the tree T.

However, when linear tests are considered, it is necessary to change the penalty for tree over-parametrization. It is rather straightforward that an oblique split based on few features is more complex than a univariate test. As a consequence, the tree complexity should not only reflect the tree size and the number of features that constitute models in the leaves, but also the complexity of the tests in the internal nodes. However, it is not easy to arbitrarily define the balance between different measures because it depends on the problem and user preferences. In such a situation we decided to define the tree complexity $k(T)$ in most flexible way to allow the user to tune its final form:

$$k(T) = \alpha * Q(T) + \beta * M(T) + \gamma * F(T), \tag{6}$$

where $Q(T)$ is the number of nodes in model tree T and $M(T)$ is the sum of the number of all features in the linear models at the leaves. Additional penalty term $F(T)$ works only for oblique trees as it sums up the number of features in multivariate tests. Default values of the parameters are: $\alpha = 2.0$, $\beta = 1.0$ and $\gamma = 1.0$, however further research to determine theirs values should be considered.

2.5 Selection and Termination Condition

Ranking linear selection [20] is applied as a selection mechanism. Additionally, in each iteration a single individual with the highest value of fitness function in current population in copied to the next one *(elitist strategy)*. The evolution terminates when the fitness of the best individual in the population does not improve during the fixed number of generations. In case of a slow convergence, maximum number of generations is also specified, which allows us to limit computation time.

2.6 Genetic Operators

Genetic operators are the two main forces that form the basis of evolutionary systems and provide a necessary diversity and novelty. Tree-based representation requires developing specialized operators corresponding to the classical mutation and cross-over. Previously performed research [15] shows that the recombination of two individuals usually has much higher destructive power and context change than the mutation, therefore the default probability to select cross-over is equal 0.2 and mutation 0.8.

Cross-over solution starts with selecting positions in two affected individuals. We propose three variants of recombination [6] that involve exchanging oblique tests in internal nodes, subtrees and branches between the nodes of individuals.

Mutation solution starts with randomly choosing the type of node (equal probability to select leaf or internal node). Next, the ranked list of nodes of the selected type is created and a mechanism analogous to ranking linear selection is applied to decide which node will be affected. Depending on the type of node, ranking takes into account the location of the internal node (internal nodes in lower parts of the tree are mutated with higher probability) and the absolute error (worse in terms of prediction accuracy leaves and internal nodes are mutated with higher probability). Previously [5,6] we have proposed several variants of mutation for univariate tests in internal nodes, models in the leaves and modifications in the tree structure (pruning the internal nodes and expanding the leaves). In this paper, we present new mutation operators for modifying oblique splits in internal nodes:

- test can be replaced by a new, effective one which is searched based on randomly chosen *long dipole*;
- hyper-plane can be modified by changing one, randomly chosen weight w_i;
- hyper-plane is shifted in such a way that it divides new, randomly chosen *long dipole*.

3 Preliminary Experimental Results

The proposed approach is evaluated on both artificial and real life datasets. It is compared only to the univariate versions of our global inducers: univariate Global Regression Tree (*uGRT*) [16] and univariate Global Model Tree (*uGMT*) [6], since in previous papers [5,6] we compared our solutions with popular counterparts. All presented results correspond to average of 10 runs. The Root mean squared error (*RMSE*) is given as the prediction error measure of the tested systems. Depending on the tree type, complexity measure can cover the tree size (*size*) and average number of features in the leaves (*model*).

The performance of proposed solution is compared on eight artificially generated datasets illustrated in Figure 1. They are variants of the datasets *split plane* and *armchair*. Each dataset can be perfectly predicted with regression trees (univariate a and b; oblique c and d) and model trees (univariate e and f; oblique g and h). In addition, all algorithms are tested on real-life datasets from UCI Machine Learning Repository [3].

Table 1 presents characteristics of investigated datasets and obtained results. Experiments performed on artificial datasets show the importance of tree representation in solving different prediction problems. Proposed solution *oGMT* has the most advanced splits in internal and the models in the leaves, therefore theoretically it is capable to perfectly predict values for each dataset. The major drawback of *oGMT* is that it requires much more generations to find good solution. In specified maximum number of generations (default: 100000) algorithm did not found optimal decisions for each of the 10 runs. Rest of the algorithms managed to find good decisions only when the tree representation was capable to do it.

Figure 2 illustrates *RMSE* and tree size for the best individual so far in evolution, for each tested algorithm on dataset *armchair (b)* and *armchair (h)*. When the data can be perfectly predicted with univariate regression trees (*armchair (b)*), *uGRT* and *uGMT* find optimal decisions almost instantly. Trees with oblique tests in internal nodes need much more iterations to perfectly predict data (over 1000) and even more to find the best tree structure. On the other side, when the *armchair (h)* with non-axis parallel decision border was investigated, application of algorithms with univariate tests *uGRT* and *uGMT* lead to their approximation by a very complicated stair-like structure. Similar situation was for the regression tree *oGRT* when it tried to predict linear regression models.

Presented in Table 1 differences between algorithms on the real-life datasets are not so visible. In general, the main benefit of application oblique tests instead of univariate is that the generated trees are much smaller. This applies to both regression and model trees. Preliminary results suggest also that the application of oblique tests may improve the prediction performance of *GRT* and *GMT* algorithms however the main drawback is the calculation time. In addition, there are two possible reasons why oblique trees have slightly lower than expected *RMSE* results on $Auto - Mpg$ dataset (*uGRT* vs *oGRT*) and *Pyrimidines* dataset (*uGMT* vs *oGMT*). The first plausible reason is the very slow convergence of the oblique regression and model trees. Because of the much

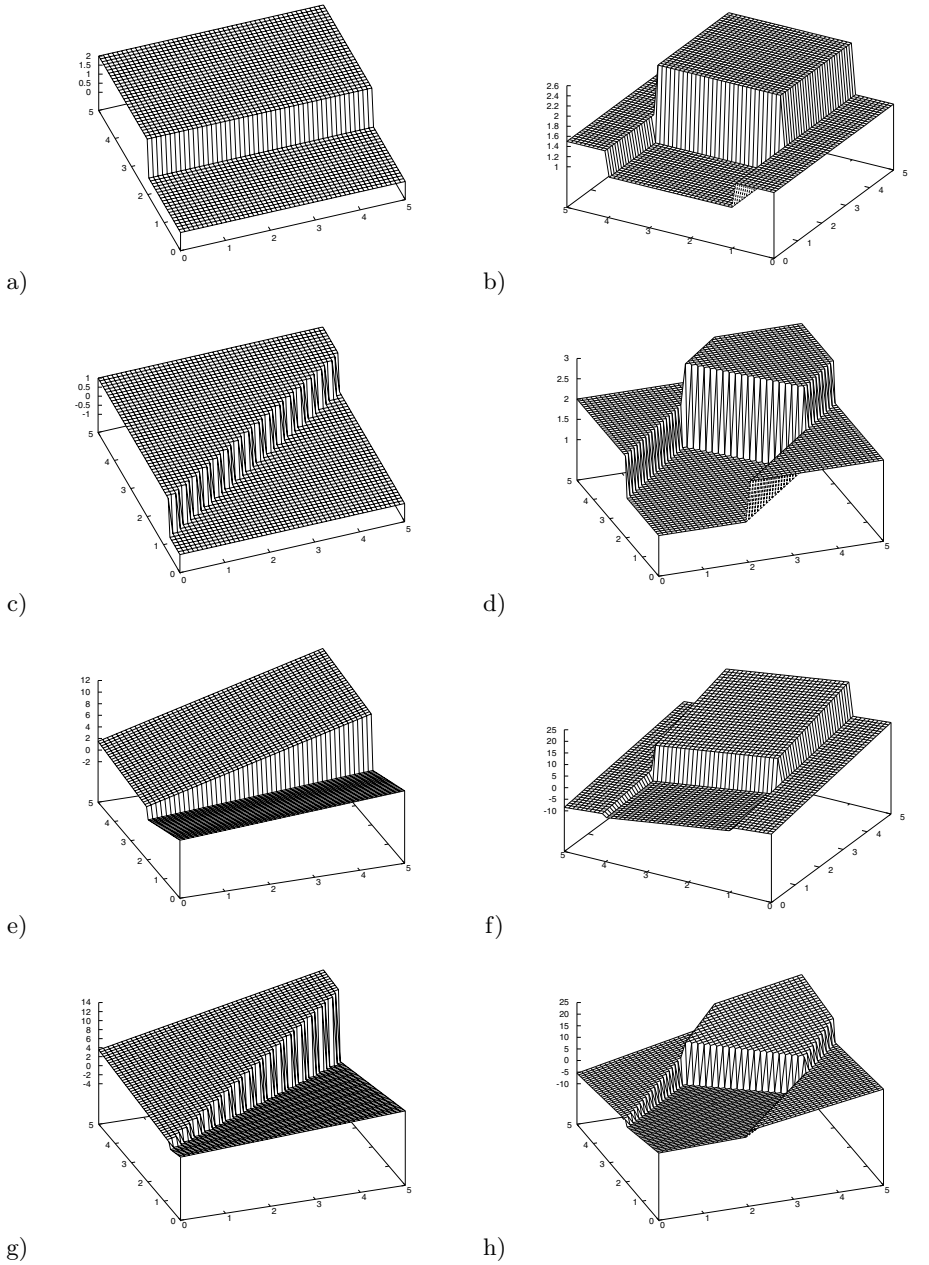


Fig. 1. Examples of artificial datasets (*split plane* - left, *armchair* - right) that can be perfectly predicted with univariate (*a*, *b*) and oblique (*c*, *d*) regression trees and univariate (*e*, *f*) and oblique (*g*, *h*) model trees

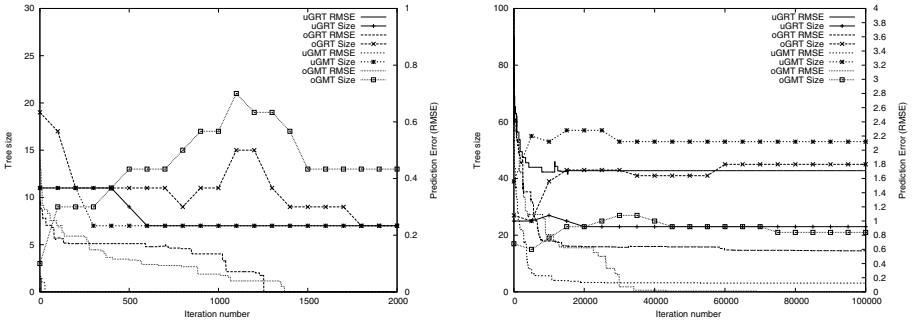


Fig. 2. Influence of the tree representation on performance of the best individual on *armchair b* (left image) and *armchair h* (right image) dataset

larger solution space the evolution could have been terminated too quickly. Second reason may refer to overfitting the oblique trees to the training data. This requires further research to determine the penalty term for over-parametrization in the fitness function.

Table 1. Characteristics of the artificial and real-life datasets (number of objects/number of numeric features/number of nominal features) and obtained results for regression trees: univariate *uGRT* and oblique *oGRT*; and for model trees: univariate *uGMT* and oblique *oGMT*

		uGRT		*oGRT*		*uGMT*			*oGMT*		
Dataset	Properties	RMSE	size	RMSE	size	RMSE	size	model	RMSE	size	model
Split plane (a)	1000/2/0	0.044	2.0	0.080	2.0	0.039	2.0	0.0	0.083	2.0	0.0
Split plane (c)	1000/2/0	0.298	13.2	0.084	2.0	0.304	14.0	0.0	0.065	2.0	0.0
Split plane (e)	1000/2/0	0.369	15.8	1.026	22.2	0.159	2.0	4.0	0.322	2.0	4.0
Split plane (g)	1000/2/0	1.369	22.6	1.093	19.9	1.380	13.4	21.6	0.612	2.2	4.2
Armchair (b)	1000/2/0	0.068	4.0	0.100	4.0	0.068	4.0	0.0	0.100	4.0	0.0
Armchair (d)	1000/2/0	0.262	17.7	0.186	5.2	0.254	18.4	0.0	0.202	6.7	0.0
Armchair (f)	1000/2/0	1.320	20.6	2.897	21.6	0.913	4.0	8.0	0.881	4.1	8.2
Armchair (h)	1000/2/0	2.320	21.5	2.539	27.9	2.019	24.5	29.7	1.300	7.7	13.0
Auto-Mpg	392/4/3	3.212	10.1	3.373	5.4	3.211	3.8	10.2	2.997	2.0	6.8
Pyrimidines	72/27/0	0.108	4.6	0.088	4.5	0.102	2.1	14.1	0.103	2.1	11.8
Triazines	186/60/0	0.146	3.8	0.140	3.1	0.144	2.4	13.7	0.138	2.1	10.7

4 Conclusion and Future Works

In the paper the global induction of oblique model trees was investigated. In contrast to classical top-down inducers, where locally optimal tests are sequentially chosen, in *GMT* the tree structure, oblique tests in internal nodes and models in the leaves are searched in the same time by specialized evolutionary algorithm. Preliminary experimental results show that the globally evolved

oblique regression and model trees are highly competitive compared to univariate counterparts. Extending the representation of tests in internal nodes open new possibilities in finding better predictions.

Proposed approach is constantly improved. Further research to determine more appropriate value of complexity penalty term in the *BIC* criterion is advised and other commonly used measures should be considered. Currently we are working on a *mixed GMT* that will be able to self-adapt structure of the tree: appropriate test in internal node (univariate or oblique) and leaf type (regression or model). On the other hand, we plan to parallelize the evolutionary algorithm and add local optimizations in order to speed-up and focus evolutionary search.

Acknowledgments. This work was supported by the grant S/WI/2/13 from Bialystok University of Technology.

References

1. Barros, R.C., Ruiz, D.D., Basgalupp, M.: Evolutionary model trees for handling continuous classes in machine learning. Information Sciences 181, 954–971 (2011)
2. Barros, R.C., Basgalupp, M.P., Carvalho, A.C., Freitas, A.A.: A Survey of Evolutionary Algorithms for Decision-Tree Induction. IEEE Transactions on Systems Man and Cybernetics, Part C 42(3), 291–312 (2012)
3. Blake, C., Keogh, E., Merz, C.: UCI Repository of Machine Learning Databases (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
4. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth Int. Group (1984)
5. Czajkowski, M., Kretowski, M.: Globally Induced Model Trees: An Evolutionary Approach. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6238, pp. 324–333. Springer, Heidelberg (2010)
6. Czajkowski, M., Kretowski, M.: An Evolutionary Algorithm for Global Induction of Regression Trees with Multivariate Linear Models. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS, vol. 6804, pp. 230–239. Springer, Heidelberg (2011)
7. Czajkowski, M., Kretowski, M.: Does Memetic Approach Improve Global Induction of Regression and Model Trees? In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) EC 2012 and SIDE 2012. LNCS, vol. 7269, pp. 174–181. Springer, Heidelberg (2012)
8. Dobra, A., Gehrke, J.: SECRET: A Scalable Linear Regression Tree Algorithm. In: Proc. of KDD (2002)
9. Fan, G., Gray, J.B.: Regression tree analysis using target. Journal of Computational and Graphical Statistics 14(1), 206–218 (2005)
10. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.): Advances in Knowledge Discovery and Data Mining. AAAI Press (1996)
11. Gagne, P., Dayton, C.M.: Best Regression Model Using Information Criteria. Journal of Modern Applied Statistical Methods 1, 479–488 (2002)
12. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Data Mining, Inference, and Prediction, 2nd edn. Springer (2009)
13. Karalic, A.: Linear Regression in Regression Tree Leaves. International School for Synthesis of Expert Knowledge, Bled, Slovenia (1992)

14. Kotsiantis, S.B.: Decision trees: a recent overview. Artificial Intelligence Review, 1–23 (2011)
15. Kretowski, M., Grzes, M.: Global induction of oblique decision trees: An evolutionary approach, Intelligent Information Processing and Web Mining. In: Proc. of the IIS. Advances in Soft Computing, pp. 309–318 (2005)
16. Kretowski, M., Czajkowski, M.: An Evolutionary Algorithm for Global Induction of Regression Trees. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS (LNAI), vol. 6114, pp. 157–164. Springer, Heidelberg (2010)
17. Li, K.C., Lue, H.H., Chen, C.H.: Interactive Tree-Structured Regression via Principal Hessian Directions. Journal of the American Statistical Association 95, 547–560 (2000)
18. Llorà, X., Wilson, S.W.: Mixed decision trees: Minimizing knowledge representation bias in LCS. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3103, pp. 797–809. Springer, Heidelberg (2004)
19. Malerba, D., Esposito, F., Ceci, M., Appice, A.: Top-down Induction of Model Trees with Regression and Splitting Nodes. IEEE Trans. on PAMI 26(5), 612–625 (2004)
20. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 3rd edn. Springer (1996)
21. Naumov, G.E.: NP-completeness of problems of construction of optimal decision trees. Soviet Physics Doklady 36(4), 270–271 (1991)
22. Potts, D., Sammut, C.: Incremental Learning of Linear Model Trees. Machine Learning 62, 5–48 (2005)
23. Potgieter, G., Engelbrecht, A.: Evolving model trees for mining data sets with continuous-valued classes. Expert Systems with Applications 35, 1513–1532 (2008)
24. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical Recipes in C. Cambridge University Press (1988)
25. Rokach, L., Maimon, O.: Top-down induction of decision trees classifiers - A survey. IEEE Transactions on Systems, Man, and Cybernetics - Part C 35(4), 476–487 (2005)
26. Rokach, L., Maimon, O.Z.: Data mining with decision trees: theory and application. Machine Perception Arfitical Intelligence 69 (2008)
27. Schwarz, G.: Estimating the Dimension of a Model. The Annals of Statistics 6, 461–464 (1978)
28. Torgo, L.: Functional Models for Regression Tree Leaves. In: Proc. of ICML, pp. 385–393. Morgan Kaufmann (1997)
29. Quinlan, J.: Learning with Continuous Classes. In: Proc. of AI 1992, pp. 343–348. World Scientific (1992)
30. Vogel, D., Asparouhov, O., Scheffer, T.: Scalable look-ahead linear regression trees. In: Proc. of 13th ACM SIGKDD, pp. 757–764. ACM Press, New York (2007)

Genetic Cost Optimization of the $GI/M/1/N$ Finite-Buffer Queue with a Single Vacation Policy

Marcin Gabryel[1], Robert K. Nowicki[1],
Marcin Woźniak[2], and Wojciech M. Kempa[2]

[1] Institute of Computational Intelligence, Czestochowa University of Technology,
Al. Armii Krajowej 36, 42-200 Czestochowa, Poland
[2] Institute of Mathematics, Silesian University of Technology,
ul. Kaszubska 23, 44-100 Gliwice, Poland
{Marcin.Gabryel,Robert.Nowicki}@iisi.pcz.pl,
{Marcin.Wozniak,Wojciech.Kempa}@polsl.pl

Abstract. In the artice, problem of the cost optimization of the $GI/M/1/N$-type queue with finite buffer and a single vacation policy is analyzed. Basing on the explicit representation for the joint transform of the first busy period, first idle time and the number of packets transmitted during the first busy period and fixed values of unit costs of the server's functioning an optimal set of system parameters is found for exponentially distributed vacation period and 2-Erlang distribution of inter arrival times. The problem of optimization is solved using genetic algorithm. Different variants of the load of the system are considered as well.

Keywords: Busy period, finite-buffer queue, genetic algorithm, idle time, optimization, single vacation.

1 Introduction

Finite-buffer queues are being intensively studied nowadays due to their numerous applications in different real-life problems occurring in technical and economical sciences. They are used, first of all, in modelling of telecommunication and computer networks (ATM switches, IP routers etc.), however they can also be helpful in the investigation of manufacturing processes and in some problems occurring in logistics and transport.

From the observation follows that the stream of IP packets incoming to the Internet router or cells arriving into the ATM switch rather rarely can be described by a Poisson process. Moreover, due to permanent changing intensity of the traffic and some sophisticated phenomena characterizing the traffic, like the self-similarity or burstiness, the "classical" analysis limited to the stationary state of the system, is not sufficient for a thorough description of the system's operation. These arguments motivate the transient analysis of the system's operation basing on the non-Poisson process describing the input flow of packets.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 12–23, 2013.

An extension of the class of finite-buffer queues are models with server's vacation, where the server takes a randomly distributed time during which the service process is blocked. Finite systems with server vacations can be used in modelling of SVC (switched virtual connection) networks, where the vacation period can be considered as a time needed for the server to release one SVC or the time for setting up any next SVC (see [11]).

In the paper we investigate the $GI/M/1$-type finite-buffer queuing system with single vacation policy and exhaustive service. Basing on the explicit representation for the joint transform of the first busy period, first idle time and the number of packets transmitted during the first busy period, and fixed values of unit costs of the server's functioning during the service and being idle, we find an optimal set of system parameters for exponentially distributed vacation period and 2-Erlang distribution of inter arrival times. In fact, we obtain results for different variants of the load of the system. The problem of optimization is solved using genetic algorithms.

Results for vacation queuing models can be found in monographs [14], [18] and e.g. in papers [2], [3], [11] and [12] and [15]. A batch-arrival $GI/G/1$-type system with infinite buffer and exponentially distributed single vacations is considered in [6] on the first vacation cycle, using the approach based on Wiener-Hopf-type factorization. Transient characteristics of the system with single vacations with Poisson arrivals and generally distributed service times are analyzed in [7], [8] and [9].

In [16] the classical linear cost structure is introduced in the queuing model. The problem of cost optimization is also investigated e.g. in [4]. In [10] the problem of the existence of the optimal vacation policy is solved. In [17] the formula for the total expected cost per unit time in the finite-buffer $M/G/1$-type system with removable server, working in the stationary regime, is obtained.

This paper presents an innovative application of genetic algorithms for positioning queuing systems. The authors of [1] and [19] presented other interesting applications of genetic algorithms to simulate examined objects. Built genetic simulation system helped to calculate optimal cost for modeled queuing system. Presented research describe system positioning in some possible scenarios, but described genetic method makes it also possible evaluate optimal values of system variables in other conditions.

1.1 Queuing Model

In the paper we study a finite-buffer $GI/M/1/N$-type queuing system with inter arrival times with generally distributed random variables with a distribution function $F(\cdot)$, and exponential service times with mean μ^{-1}. The system capacity equals N i.e. we have the $(N-1)$-places in buffer and one place in service. We assume that the system starts working at $t = 0$ with at least one packet present. After each busy period the server takes a single vacation, with a general type distribution function $V(\cdot)$, during which the service process is stopped. If at the end of the vacation there is no packet in the buffer queue, the server is being activated (is on standby) and waits for the first arrival to start the service process.

If the system is not empty at the vacation completion epoch, then the service process is being initialized immediately and the new busy period begins.

We assume that sequences of successive inter arrival times, service times and the single vacation duration are totally independent random variables.

Let us introduce the following notations:

- τ_1 — the first busy period of the system (starting at $t = 0$);
- δ_1 — the first idle time of the system (consisting of the first vacation time v_1 and the first server standby time q_1);
- $h(\tau_1)$ — the number of packets completely served during τ_1;
- $X(t)$ — the number of packets present in the system at time t.

1.2 Auxiliary Results

In [13] the explicit formula for the conditional joint characteristic function of random variables τ_1, δ_1 and $h(\tau_1)$ is derived.

Define

$$B_n(s, \varrho, z) = \mathbf{E}\{e^{-s\tau_1 - \varrho\delta_1} z^{h(\tau_1)} \mid X(0) = n\}, \tag{1}$$

where $1 \leq n \leq N$, $s \geq 0$, $\varrho \geq 0$ and $|z| \leq 1$.

Introduce the following functions:

$$f(s) = \int_0^\infty e^{-st} dF(t), \quad s > 0 \tag{2}$$

and besides

$$a_n(s, z) = \int_0^\infty \frac{(z\mu t)^n}{n!} e^{-(\mu+s)t} dF(t), \quad n \geq 0, \tag{3}$$

$$\Psi_n(s, \varrho, z) = -\frac{(z\mu)^n}{(n-1)!}\Big[\int_0^\infty dF(t) \int_0^t x^{n-1} e^{-(\mu+s)x} \times \left(e^{-\varrho(t-x)} V(t-x) + \int_{t-x}^\infty e^{-\varrho y} dV(y)\right) dx\Big]. \tag{4}$$

Moreover, basing on the sequence $\big(a_n(s, z)\big)$ defined in Eq. (3), we defined recursively the following sequence:

$$R_0(s, z) = 0, \quad R_1(s, z) = a_0^{-1}(s, z),$$
$$R_{n+1}(s, z) = R_1(s, z)(R_n(s, z) - \sum_{k=0}^{n} a_{k+1}(s, z) R_{n-k}(s, z)). \tag{5}$$

Finally, let

$$D(s, \varrho, z) = \sum_{k=1}^{N-1} a_k(s, z) \sum_{i=2}^{N-k+1} R_{N-k+1-i}(s, z) \Psi_i(s, \varrho, z), \tag{6}$$

$$G(s, \varrho, z) = \Psi_N(s, \varrho, z) + \big(1 - f(\mu + s)\big) \sum_{k=2}^{N} R_{N-k}(s, z) \Psi_k(s, \varrho, z) \tag{7}$$

and

$$H(s,z) = \big(1 - f(\mu+s)\big)R_{N-1}(s,z) - \sum_{k=1}^{N-1} a_k(s,z)R_{N-k}(s,z). \tag{8}$$

The following theorem is true.

Theorem 1. *For $s \geq 0$, $\varrho \geq 0$ and $|z| \leq 1$ the following formulae hold true:*

$$\begin{aligned} B_1(s,\varrho,z) &= \mathbf{E}\{e^{-s\tau_1-\varrho\delta_1} z^{h(\tau_1)} \mid X(0)=1\} \\ &= \frac{D(s,\varrho,z) - G(s,\varrho,z)}{H(s,z)} - \Psi_1(s,\varrho,z) \end{aligned} \tag{9}$$

and

$$\begin{aligned} B_n(s,\varrho,z) &= \mathbf{E}\{e^{-s\tau_1-\varrho\delta_1} z^{h(\tau_1)} \mid X(0)=n\} \\ &= \frac{D(s,\varrho,z) - G(s,\varrho,z)}{H(s,z)} R_{n-1}(s,z) - \sum_{k=2}^{n} R_{n-k}(s,z)\Psi_k(s,\varrho,z), \quad 2 \leq n \leq N. \end{aligned} \tag{10}$$

where $n \geq 1$ and $a_n(s,z)$, $\Psi_n(s,\varrho,z)$, $D(s,\varrho,z)$, $G(s,\varrho,z)$ and $H(s,z)$ are defined in (3), (4), (6), (7) and (8) respectively.

1.3 Cost Optimization Problem

In the paper we are interested in the solution of the following optimization problem. Consider the following equation

$$Q_n(c_1) = r(\tau_1)\mathbf{E}_n\tau_1 + r(\delta_1)\mathbf{E}_n\delta_1, \tag{11}$$

in which $r(\tau_1)$ and $r(\delta_1)$ denote, respectively, the fixed unit costs of the system's operation during the first busy period τ_1 and the first idle time δ_1, $\mathbf{E}_n\tau_1$ and $\mathbf{E}_n\delta_1$ are, respectively, the means of the first busy period τ_1 and the first idle time δ_1 on condition that the system starts evolution with n packets present. Besides, $Q_n(c_1)$ is the total cost of the operation of the system during the first cycle c_1 on condition that $X(0) = n$.

Of course the unit cost of the operation of the system during the first cycle c_1, on condition that the system contains exactly n packets at the opening, one can expressed as

$$r_n(c_1) = \frac{Q_n(c_1)}{\mathbf{E}_n(c_1)} = \frac{r(\tau_1)\mathbf{E}_n\tau_1 + r(\delta_1)\mathbf{E}_n\delta_1}{\mathbf{E}_n\tau_1 + \mathbf{E}_n\delta_1}. \tag{12}$$

Let us consider the system in which inter arrival times are exponentially distributed and the vacation period has 2-Erlang distribution. Namely, let us take

$$F(t) = 1 - e^{-\lambda t}, \quad \lambda > 0,\, t > 0, \tag{13}$$

and

$$V(t) = 1 - e^{-\alpha t}(1 + \alpha t), \quad \alpha > 0, t > 0. \tag{14}$$

For such a system we use genetic optimization for finding the optimal set of "input" system parameters λ, μ and α, to minimize the costs $Q_n(c_1)$ and/or $r_n(c_1)$. In fact we will take into consideration different variants of the load of the system: the case of under-load, critically load and overload.

1.4 Genetic Cost Optimization

Very important phase in the process of designing queuing systems is positioning for optimal costs of work. To perform this operation we must know possible malfunctions and optimal work conditions. To do this we may apply the knowledge that comes from human experts or our previous experience. In many projects this is a main source which determines the prototypes. However, usually it is insufficient. The best way is to test the system in practice or perform computer simulations. First way may be vary long in time and cost a lot. Therefore best solution is to use computer. The question is, which method to use? We would like to present positioning of queuing system by the use of genetic algorithm. Computational intelligence, especially genetic algorithms, are very effective. In the literature [1] and [19] the authors described use of genetic algorithm to create knowledge about technical systems. Presented use of computational intelligence help to simulate the object states and build decision support systems. In this paper we present possible way to use genetic to calculate and position queuing system. However queuing systems have complicated mathematical models therefore genetic calculation is still not easy. We present simulation results of $GI/M/1/N$ finite-buffer queue with a single vacation policy. We used standard genetic algorithm to built a dedicated evolutionary simulation system based on mathematical model described in Section 1.2. In the research we used formula (12) to optimize cost of the system. Genetic simulation system was searching for best values of $GI/M/1/N$ queuing system variables that make it work with lowest costs in the specified time unit. The research results provide type of knowledge that describes examples of proper operation of the system in some possible scenarios. This type of knowledge is necessary for its tuning and evaluating.

2 Research Results

In this section we will present research results for optimum values of the examined $GI/M/1/N$ queue system. Based on the research results, we may predict possible time of system response in each case and optimize cost of work $r_n(c_1)$, described in Eq. (12). In the research, we have examined the system for optimum values in 100 samplings. Presented optimal results are average values. We will determine time prediction based on the following assumptions:

- Average service time: $T_{service} = \frac{1}{\mu}$,

- Average time between packages income into the system: $T_{income} = \frac{2}{\lambda}$,
- Average vacation time: $T_{vacation} = \frac{1}{\alpha}$,
- Examined system size: N = buffer size +1.

In Table 1 we present optimum values for all system parameters μ, λ and α. However in the real environment we are able to set some values of the sys-

Table 1. Optimimal parameters μ, λ and α for lowest value of Eq.(12)

	μ	λ	α
Average optimal value	2.104179284	0.049898874	13092.21989
optimal $r_n(c_1)$	0.011857087		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.475244675	40.08106516	$7.63812E^{-05}$

tem we use. Therefore we have also tried to optimized values of parameters μ, λ and α in few possible scenarios. Each scenario was defined and then, there were optimized values of system parameters and cost of work according to given assumptions. In each scenario there were 100 genetic optimization experiments and the results are given as average value of optimization for all the experiments.

Scenario 1
In this scenario, to genetic cost optimization we have assumed that the system handles incoming packets in a constant time what means that, average service time: $T_{service} = \frac{1}{\mu}$ is constant. Therefore we have set the value of μ parameter and optimized other system parameters. In this scenario all system parameters were optimized for set values: $\mu = 100$, $\mu = 1$ or $\mu = 0.01$. Research results are shown in Table 2.

Scenario 2
In this scenario, to genetic cost optimization we have assumed that packets come into the system with some regularity, time between packages income into the system: $T_{income} = \frac{2}{\lambda}$ is constant. Therefore we have set the value of λ parameter and optimized other system parameters. In this scenario all system parameters were optimized for set values: $\lambda = 100$, $\lambda = 1$ or $\lambda = 0.01$. Research results are shown in Table 3.

Scenario 3
In this scenario, to genetic cost optimization we have assumed that system need to stop serving packets with some regularity, vacation time: $T_{vacation} = \frac{1}{\alpha}$ is constant. Therefore we have set the value of α parameter and optimized other system parameters. In this scenario all system parameters were optimized for set values: $\alpha = 100$, $\alpha = 1$ or $\alpha = 0.01$. Research results are shown in Table 4. Moreover we have also analyzed some more complicated scenarios. In the research were examined possible situations where service time, packets income time or vacation time is set and cost of system work must be adequate.

Table 2. Optimal parameters λ and α for set $\mu = 100$, $\mu = 1$, $\mu = 0.01$ and lowest cost value of Eq.(12)

	μ	λ	α
Average optimal value	100	0.441648853	1.092121241
optimal $r_n(c_1)$	0.002208244		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.01	4.528484533	0.915649255
Average optimal value	1	0.185442954	1.456477569
optimal $r_n(c_1)$	0.092721477		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	1	10.78498781	0.686587985
Average optimal value	0.01	0.258147693	1.594055468
optimal $r_n(c_1)$	12.90738467		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	100	7.747502887	0.627330742

Table 3. Optimal parameters μ and α for set $\lambda = 100$, $\lambda = 1$, $\lambda = 0.01$ and lowest cost value of Eq.(12)

	μ	λ	α
Average optimal value	4.059353298	100	$2.3177E^{-06}$
optimal $r_n(c_1)$	12.3172329		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.246344658	0.02	431462.2255
Average optimal value	27.30386864	1	0.000002924
optimal $r_n(c_1)$	0.018312423		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.036624847	2	341997.264
Average optimal value	1.356968521	0.01	1.503990611
optimal $r_n(c_1)$	0.003684684		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.736936771	200	0.664897768

Scenario 4

In this scenario, to genetic cost optimization we have assumed, similar to scenario 2, that service time: $T_{service} = \frac{1}{\mu}$ is constant. Therefore we have set the value of μ parameter and optimized other system parameters. Moreover we have assumed that cost of system work is $r_n(c_1)$ is also defined in some way. In this scenario all system parameters were optimized for set values: $\mu = 100$, $\mu = 1$ or $\mu = 0.01$ and $r_n(c_1) < 1$ or $r_n(c_1) > 1$. Research results are shown in Table 5 - Table 6. We have also optimized cost of the system for set time of packets income an vacation.

Table 4. Optimal parameters μ and λ for set $\alpha = 100$, $\alpha = 1$, $\alpha = 0.01$ and lowest cost value of Eq.(12)

	μ	λ	α
Average optimal value	1.16928266	0.177362545	100
optimal $r_n(c_1)$	0.075842459		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.855225203	11.27633797	0.01
Average optimal value	1.247463554	0.172708087	1
optimal $r_n(c_1)$	0.069223701		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.801626626	11.58023365	1
Average optimal value	1.227674838	0.206532185	0.01
optimal $r_n(c_1)$	0.084115182		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.814547931	9.683720705	100

Table 5. Optimal parameters α and λ for set μ and lowest cost value of Eq.(12)< 1

	μ	λ	α
Average optimal value	100	0.400070456	1.192386163
optimal $r_n(c_1) < 1$	0.002000352		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.01	4.999119455	0.838654482
Average optimal value	1	0.223777444	1.300852544
optimal $r_n(c_1) < 1$	0.111888722		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	1	8.937451265	0.768726636
Average optimal value	0.01	0.008013686	7.000048983
optimal $r_n(c_1) < 1$	0.4006843		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	100	249.5730429	0.142856143

Scenario 5

In this scenario, to genetic cost optimization we have assumed, similar to scenario 2, that time between packages income into the system: $T_{income} = \frac{2}{\lambda}$ is constant. Therefore we have set the value of λ parameter and optimized other system parameters. Moreover we have assumed that cost of system work $r_n(c_1)$ is also defined in some way. In this scenario all system parameters were optimized for set values: $\lambda = 100$, $\lambda = 1$ or $\lambda = 0.01$ and $r_n(c_1) < 1$ or $r_n(c_1) > 1$. Research results are shown in Table 7 - Teble 8. Last scenario present research results on optimized cost of the system for set vacation time.

Table 6. Optimal parameters α and λ for set μ and lowest cost value of Eq.(12)> 1

	μ	λ	α
Average optimal value	100	1310.416361	0.000056554
optimal $r_n(c_1) > 1$	6.552081805		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.01	0.001526232	17682.21523
Average optimal value	1	16.55791579	0.000000459
optimal $r_n(c_1) > 1$	8.278957897		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	1	0.120788149	2178649.237
Average optimal value	0.01	0.232515381	1.47718659
optimal $r_n(c_1) > 1$	11.62576905		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	100	8.601581501	0.676962549

Table 7. Optimal parameters α and μ for set λ and lowest cost value of Eq.(12)< 1

	μ	λ	α
Average optimal value	1598.611383	100	0.000278277
optimal $r_n(c_1) < 1$	0.031277145		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.000625543	0.02	3593.541687
Average optimal value	1.311446436	1	0.184496128
optimal $r_n(c_1) < 1$	0.070340703		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.762516846	2	10.8403359
Average optimal value	1.715058427	0.01	1.357447045
optimal $r_n(c_1) < 1$	0.002915353		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.583070515	200	0.736676988

Scenario 6

In this scenario, to genetic cost optimization we have assumed, similar to scenario 2, that vacation time: $T_{vacation} = \frac{1}{\alpha}$ is constant. Therefore we have set the value of α parameter and optimized other system parameters. Moreover we have assumed that cost of system work $r_n(c_1)$ is also defined in some way. In this scenario all system parameters were optimized for set values: $\alpha = 100$, $\alpha = 1$ or $\alpha = 0.01$ and $r_n(c_1) < 1$ or $r_n(c_1) > 1$. Research results are shown in Table 9 - Table 10.

Table 8. Optimal parameters α and μ for set λ and lowest cost value of Eq.(12)> 1

	μ	λ	α
Average optimal value	4.772254414	100	0.000003217
optimal $r_n(c_1) > 1$	10.47722851		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.20954457	0.02	310848.6167
Average optimal value	0.288431013	1	0.000006832
optimal $r_n(c_1) > 1$	1.733516777		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	3.467033554	2	146370.0234
Average optimal value	0.002060192	0.01	25.22387757
optimal $r_n(c_1) > 1$	2.426958264		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	485.3916528	200	0.039644975

Table 9. Optimal parameters λ and μ for set α and lowest cost value of Eq.(12)< 1

	μ	λ	α
Average optimal value	1.416040912	0.216560935	100
optimal $r_n(c_1) < 1$	0.076467047		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.706194285	9.235275974	0.01
Average optimal value	1.737398608	0.115104772	1
optimal $r_n(c_1) < 1$	0.033125608		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.575573156	17.37547423	1
Average optimal value	1.34126515	0.180202828	0.01
optimal $r_n(c_1) < 1$	0.067176437		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	0.745564738	11.09860496	100

3 Final Remarks

In the article, we have proposed a new method for queuing systems positioning. Genetic algorithms are useful in simulations or to generate a collection of representative samples as presented by the authors of [1] and [19], which can be used by decision support systems. They are also very effective for positioning queuing systems. This method can be useful when we have a model of positioned object and because of its complexity classical model calculations are merely feasible. Conducted experiments confirm its usefulness to simulate examined object in

Table 10. Optimal parameters λ and μ for set α and lowest cost value of Eq.(12)> 1

	μ	λ	α
Average optimal value	0.129801621	0.448907478	100
optimal $r_n(c_1) > 1$	1.729205978		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	7.704064035	4.455261046	0.01
Average optimal value	0.153689121	0.479652145	1
optimal $r_n(c_1) > 1$	1.560462256		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	6.506641417	4.169688431	1
Average optimal value	0.10366026	0.462297133	0.01
optimal $r_n(c_1) > 1$	2.229866744		
Time	$T_{service}$	T_{income}	$T_{vacation}$
[s]	9.646898435	4.326221941	100

many possible scenarios. An important restriction is only to carry out a large number of simulations to determine the best description of the simulated object. It is therefore time-consuming procedure. Further work should be carried out in the direction of reducing time-consuming solution, eg. by using some knowledge prior to generating the initial population.

References

1. Gabryel, M., Woźniak, M., Nowicki, R.K.: Creating learning sets for control systems using an evolutionary method. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) EC 2012 and SIDE 2012. LNCS, vol. 7269, pp. 206–213. Springer, Heidelberg (2012)
2. Gupta, U.C., Banik, A.D., Pathak, S.S.: Complete analysis of $MAP/G/1/N$ queue with single (multiple) vacation(s) under limited service discipline. Journal of Applied Mathematics and Stochastic Analysis 3, 353–373 (2005)
3. Gupta, U.C., Sikdar, K.: Computing queue length distributions in $MAP/G/1/N$ queue under single and multiple vacation. Appl. Math. Comput. 174(2), 1498–1525 (2006)
4. Kella, O.: Optimal control of the vacation scheme in an $M/G/1$ queue. Oper. Res. 38(4), 724–728 (1990)
5. Kempa, W.M.: $GI/G/1/\infty$ batch arrival queuing system with a single exponential vacation. Math. Method. Oper. Res. 69(1), 81–97 (2009)
6. Kempa, W.M.: Some new results for departure process in the $M^X/G/1$ queuing system with a single vacation and exhaustive service. Stoch. Anal. Appl. 28(1), 26–43 (2009)
7. Kempa, W.M.: Characteristics of vacation cycle in the batch arrival queuing system with single vacations and exhaustive service. Int. J. Appl. Math. 23(4), 747–758 (2010)

8. Kempa, W.M.: On departure process in the batch arrival queue with single vacation and setup time. Annales UMCS, Informatica 10(1), 93–102 (2010)
9. Kempa, W.M.: The virtual waiting time in a finite-buffer queue with a single vacation policy. In: Al-Begain, K., Fiems, D., Vincent, J.-M. (eds.) ASMTA 2012. LNCS, vol. 7314, pp. 47–60. Springer, Heidelberg (2012)
10. Lillo, R.E.: Optimal operating policy for an $M/G/1$ exhaustive server-vacation model. Methodol. Comput. Appl. 2(2), 153–167 (2000)
11. Niu, Z., Takahashi, Y.: A finite-capacity queue with exhaustive vacation/close-down/setup times and Markovian arrival processes. Queueing Syst. 31, 1–23 (1999)
12. Niu, Z., Shu, T., Takahashi, Y.: A vacation queue with setup and close-down times and batch Markovian arrival processes. Perform. Evaluation 54(3), 225–248 (2003)
13. Nowicki, R.K., Kempa, W.M., Woźniak, M.: Characteristics of vacation cycle in a finite-buffer queue with single vacation policy - analytical study with optimization. Int. J. Appl. Math. Comp. Sci. (submitted)
14. Takagi, H.: Queueing Analysis, vol. 1: Vacation and Priority Systems, vol. 2. Finite Systems. North-Holland, Amsterdam (1993)
15. Takagi, H.: $M/G/1/N$ queues with server vacations and exhaustive service. Oper. Res. 42(5), 926–939 (1994)
16. Teghem Jr., J.: Control of the service process in a queueing system. Eur. J. Oper. Res. 23, 141–158 (1986)
17. Teghem Jr., J.: Optimal control of a removable server in an $M/G/1$ queue with finite capacity. Eur. J. Oper. Res. 31, 358–367 (1987)
18. Tian, N., Zhang, Z.G.: Vacation queueing models. Theory and applications. Springer, New York (2006)
19. Woźniak, M.: Model of artificial intelligence for active control of selected vehicle dynamic characteristics, PhD. Thesis, Czestochowa University of Technology, Department of Mechanical Engineering and Computer Science, Czestochowa (2012)

Hybrid Genetic-Fuzzy Algorithm for Variable Selection in Spectroscopy

Telma Woerle de Lima[1], Anderson da Silva Soares[1], Clarimar José Coelho[2], Rogério Lopes Salvini[1], and Gustavo Teodoro Laureano[1]

[1] Informatics Institute, Federal University of Goiás, Brazil
[2] Computer Science Department, Pontifical University Catholic of Goiás, Brazil
anderson@inf.ufg.br

Abstract. This paper presents a hybrid multi-objective genetic fuzzy algorithm for the variable-selection problem in spectroscopy. The problem formulation considers three fitness functions related to linear equations system stability. These fitness functions are models with fuzzy sets that evaluate the fitness solution for pick out the best to crossover. The population diversity is obtained applying the crowding distance method. The study shows that the selection by a fuzzy decision has better results than the selection by non-domination in problems where the fitness weighing is more proper than no-domination solutions.

Keywords: Multi-objective Genetic Algorithm, Fuzzy logic, Variable Selection.

1 Introduction

Spectroscopy is the study of the interaction between matter and radiated energy [8]. It is often used in physical and analytical chemistry for the identification of substances through the spectrum emitted from or absorbed by them. An example is the spectroscopic technique used to assess the concentration or amount of a given chemical (atomic, molecular, or ionic) species. In this case, the instrument that performs such measurements is a spectrometer, spectrophotometer or spectrograph.

The Figure 1 shows the absorption spectroscopy process that measures the absorption of radiation with frequency q and the intensity of absorption is given by

$$x(\lambda) = \log \frac{P_0(\lambda)}{P(\lambda)} \tag{1}$$

where $P_0(\lambda)$ is the radiation issued by the equipment and $P(\lambda)$ is the radiation issued by the sample in the wavelength λ.

The result of interaction between matter and radiated energy provides evidence of matter composition [4]. In the ideal case, the absorbance concentration inside the matter responds to one wavelength only λ_1. In practice, the concentration is connected to more than one overlapping wavelength as shown in Figure 2.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 24–35, 2013.

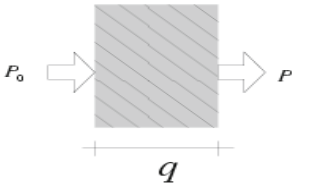


Fig. 1. Absorption spectroscopy process

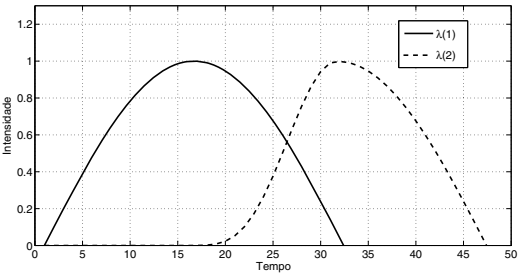


Fig. 2. Example of waves overlapping

In Figure 2 two wavelengths are shown, $\lambda(1)$ and $\lambda(2)$, that pass simultaneously in parts of space. When two or more waves are overlapping, there is a mutual perturbation in the waves. In the algebra terms the waves overlapping means high correlation among variables and can induce to mathematical problems in the calibration model process [6].

Let a samples including two absorbance matters (A and B) with overlapping spectral (Figure 2) is possible to get y_A and y_B like as

$$\begin{aligned} x(\lambda_1) &= k_A(\lambda_1)y_A + k_B(\lambda_1)y_B \\ x(\lambda_2) &= k_A(\lambda_2)y_A + k_B(\lambda_2)y_B \end{aligned} \tag{2}$$

$$\begin{aligned} \begin{bmatrix} x(\lambda_1) \\ x(\lambda_2) \end{bmatrix} &= \begin{bmatrix} k_A(\lambda_1)\ k_B(\lambda_1) \\ k_A(\lambda_2)\ k_B(\lambda_2) \end{bmatrix} \begin{bmatrix} y_A \\ y_B \end{bmatrix} \\ \begin{bmatrix} y_A \\ y_B \end{bmatrix} &= \begin{bmatrix} k_A(\lambda_1)\ k_B(\lambda_1) \\ k_A(\lambda_2)\ k_B(\lambda_2) \end{bmatrix}^{-1} \begin{bmatrix} x(\lambda_1) \\ x(\lambda_2) \end{bmatrix} \\ y_A &= b_A(\lambda_1)(\lambda_1) + b_A(\lambda_2)(\lambda_2) \\ y_B &= b_B(\lambda_1)(\lambda_1) + b_B(\lambda_2)(\lambda_2) \end{aligned} \tag{3}$$

In general terms, the multivariate model is given by

$$y = x_0 b_0 + x_1 b_1 + \ldots + x_{J-1} b_{J-1} + \epsilon \tag{4}$$

or in vectorial notation,

$$Y = X\beta + \epsilon \tag{5}$$

$$x = [x_0\ x_1 \ldots x_{J-1}], \beta = [b_0\ b_1 \ldots b_{J-1}]^T. \tag{6}$$

In the case of i samples are available with n wavelength, we can arrange in pairs $(x_i, y_i) \in \mathbf{R}^J \times \mathbf{R}$ like as

$$Y = \begin{bmatrix} y_1^a \\ y_2^a \\ \vdots \\ y_i^a \end{bmatrix} X = \begin{bmatrix} x_1^1(\lambda_1) & \dots & x_1^j(\lambda_n) \\ x_2^1(\lambda_1) & \dots & x_2^j(\lambda_n) \\ \vdots & \ddots & \vdots \\ x_i^1(\lambda_1) & \dots & x_i^j(\lambda_n) \end{bmatrix}, \tag{7}$$

where $x_i^j(\lambda_n)$ is the i-th sample of matter absorbance in the wavelength λ_n and y_i^a is the concentration of a in the i-th sample. Where the relation between the absorbance and concentration can be estimate by a coefficient matrix β that multiply X for to obtain $\hat{Y}$ estimate. The matrix X and Y are divided in X_{cal} and Y_{cal} for obtain the coefficient matrix β and X_{teste} and Y_{teste} are used to test the accuracy of prediction model. The coefficients β can be obtained by linear regression model according the Equation(8).

$$\beta = (X_{cal}^T X_{cal}) X_{cal}^T Y_{cal} \tag{8}$$

and $\hat{Y}$ can be estimate like as

$$\hat{Y} = X_{teste}\beta \tag{9}$$

Now, the problem is an oversized equation system. There is more wavelengths (variables) than samples (equations). One solution is the use of new variables obtained from linear transformations. Another solution is the elimination of redundant variables and colinearity by using a technique such as the genetic algorithm.

This paper proposes a hybrid system using the multi-objective genetic algorithm (MOGA) with a fuzzy system to select the best solutions obtained by MOGA. In order to preserve the population diversity, the Crowding Distance method was used. Besides, a comparison between the hybrid system and a standard MOGA that select the best solutions by the non-dominate method is proposed.

2 Genetic Algorithm for Variable Selection

The Genetic Algorithm (GA) is a heuristic search inspired on the process of natural selection. This heuristic is ordinarily used to generate solutions for optimization and search problems. GAs generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover. Each solution is represented by a chromosome in the GA population.

In order to perform an efficient heuristic search, it is important to balance two characteristics: exploration and exploitation. GAs achieve a remarkable balance between exploration and exploitation with their search operators.

GAs also has an implicit parallelism due to the independent evaluation of each individual in the population. In order to distinguish among the different solutions, the GAs evaluate them with an objective function, resulting in a fitness value.

However, some problems have more than an objective function. These problems represent the class of multi-objective optimization problems. Also, in most of the problems, the objectives are conflicting among each other. In thesecases the optimization algorithm should consider each objective as a fitness function.

Multi-objective Genetic Algorithm (MOGA) have been applied to the multi-objective problems. VEGA (Vector Evaluated Genetic Algorithm) was the first MOGA proposed by Schaffer in 1985 [7]. The main difference between MOGA and standard GAs is the selection operator. In some approaches MOGAs, such as SPEA, use the fitness value as a proportion to the dominance. And, other methods such as NPGA use the Pareto dominance to select solutions instead of the fitness value.

In the MOGA shown in this work, the chromosome is a binary string when each position shows if the variable is included (1) or not (0) in the regression. The recombination operator is the multi-points crossover. The MOGA proposed was tested using two processes of individual selection that will be described in the next sections.

2.1 Problem Model for Variable Selection

The variable selection include the search by a variable set from $Xcal$. In the sense we build three objective function as follow

Objective Function

1. Minimize the correlation among the variables of $Xcal$:

$$Min \sum_{p=1}^{j} \left(\frac{\sum_{j_1=p}^{j} (x^{j_1} - \bar{x}^{j_2})(x^{j_2} - \bar{x}^{j_2})}{\sqrt{\sum_{j_1=p}^{j} (x^{j_1} - \bar{x}^{j_1})^2} \sqrt{\sum_{j_2=p}^{j} (x^{j_2} - \bar{x}^{j_2})^2}} \right) \tag{10}$$

2. Maximize the correlation between $Xcal$ and $Ycal$:

$$Max \sum_{p=1}^{j} \left(\frac{\sum_{j_1=p}^{j} (x^{j_1} - \bar{x}^{j_1})(y^{j_2} - \bar{y}^{j_2})}{\sqrt{\sum_{j_1=p}^{j} (x^{j_1} - \bar{x}^{j_1})^2} \sqrt{\sum_{j_2=p}^{j} (y^{j_2} - \bar{y}^{j_2})^2}} \right) \tag{11}$$

3. Minimize the validation error in the test set.

$$Min \left(\frac{\sum_{p=1}^{i} (\hat{y}_p - y_p)^2}{i} \right) \tag{12}$$

where y_p is the p-simo element of Y_{teste} and $\hat{y}_p$ is the p-simo element of $\hat{Y}$ obtained by equation 9.

Objective function 1 minimizes the problem of ill-conditioned matrix. In the field of numerical analysis, the condition number of a function with respect to an argument measures the asymptotically worst case of how much the function can change in proportion to small changes in the argument. The objective function 2 maximizes the correlation between X and Y, i.e. we wish to select variables from X that explain the variance of Y. Function three minimizes the validation error after the regression.

2.2 Non-domination Selection

One difficulty for multi-objective optimization is how to determine the best solution. Goldberg, [3] proposed one procedure that order the solutions by the fitness value based on the number of solutions subdued. One solution is subdued by other solution if its value is worse in all objectives. For the solution that dominates other is assigned punctuation. The non-dominated solutions have bigger fitness and higher probability of selection.

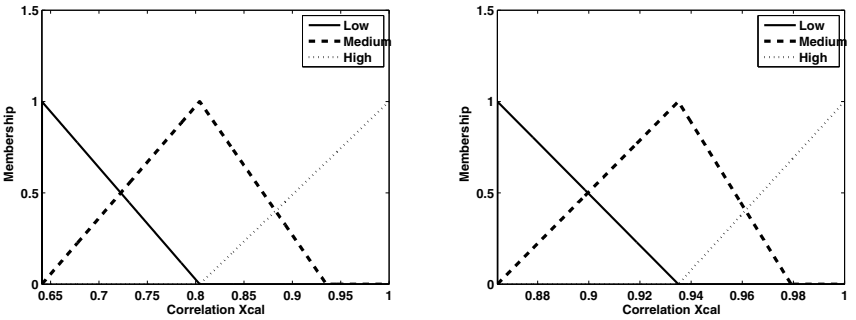
In the multi-objective optimization problem, there is no a single solution that simultaneously optimizes each objective. In that case, the objective functions are said to be conflicting, and exists a (possibly infinite number of) Pareto optimal solutions. A solution is called non-dominated, Pareto optimal, if none of the objective functions can be improved in the value without impairment of some of the other objective values [9,2].

2.3 Fuzzy System for Variable Selection

Fuzzy System is an artificial intelligent technique supported by the set and logical theory with the objective of simulating the human reasoning. The fuzzy set theory was concerned by L.A. Zadeh [10] to provide a mathematical tool for processing of approximate information. Recently, several works and applications of fuzzy systems with genetic algorithm have been proposed such as the studies of Hsiao [5] and Augugliaro [1].

The use of a fuzzy system with MOGA for the spectroscopy variable selection is proposed in this work. The fuzzy system will be employed in the step of selection the best solutions obtained from AGMO. The first input variable is the correlation in $Xcal$. Its universe of discourse is empirically defined in the values of

$Xcal$ correlation as $[(meanXcal - maxStdLines - meanStdLines) \quad maxXcal]$. This variable uses three membership function with triangular shape: *Low*, *Medium* and *High*.



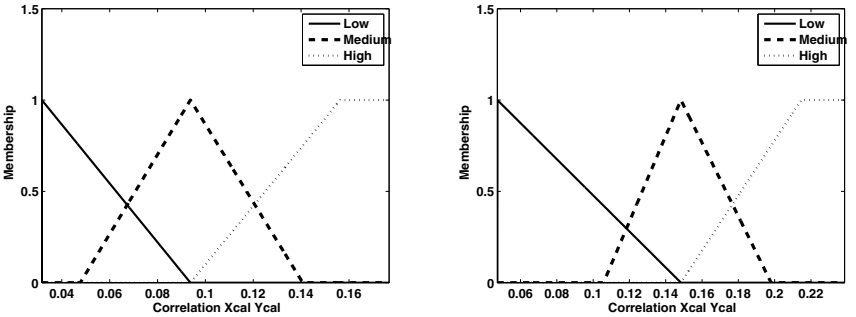
(a) Cetane concentration. (b) Sulfur concentration.

Fig. 3. Membership functions to the correlation among the variables of $Xcal$ in oil samples

For the variable correlation of $Xcal$ the points of the membership functions were defined as follows. For the function *Low*: $a = m = (meanXcal - maxStdLines - meanStdLines)$ and $b = (meanXcal - minStdLines + maxStdLines)$. For the function *Medium*: $a = (meanXcal - maxStdLines - meanStdLines)$, $m = (meanXcal - meanStdLines + maxStdLines)$ and $b = meanXcal$. And for the function *High*: $a = (meanXcal - minStdLines + maxStdLines), m = maxXcal$. Where $meanXcal$ is the mean of the correlation matrix of $Xcal$; $meanStdLines$ is the mean of the standard deviation of each line in the correlation matrix $Xcal$; $maxStdLines$ is the biggest standard deviation of the correlation matrix, $minStdLines$ is the smallest standard deviation of the lines and $maximoXcal$ is the biggest value of the correlation matrix.

Figure 3 shows the function geometric pattern of pertinence defined for a correlation objective function of $Xcal$ using the values for the wavelength of oil samples. The second input variable is the correlation between $Xcal$ and $Ycal$. It has its universe of discourse defined empirically in the values of correlation between $Xcal$ and $Ycal$ in interval $[(minXcalYcal + stdXcalYcal) \quad (meanXcalYcalAbove Par + stdXcalYcal)]$. Three membership functions for the correlations was defined: *Low*, *Middle* and *High*

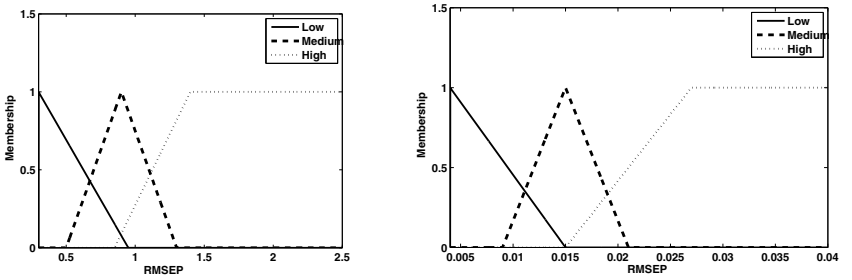
The points of the membership functions were defined as follow: for the function *Low* $a = m = (minXcalYcal + desPadXcalYcal)$ e $b = (meanXcalYcal - stdXcalYcal)$; for the function *Middle* $a = (meanXcalYcal - 1.5 * stdXcalYcal)$, $m = (meanXcalYcal - stdXcalYcal)$ e $b = (meanXcalYcal + 0.5 * stdXcalYcal)$; and for the membership function *high* $a = (meanXcalYcal - stdXcalYcal)$, $m = (meanXcalYcal + stdXcalYcal)$ e $n = b = (meanXcalYcalAbovePar + stdXcalYcal)$, where $minXcalYcal$ is the less correlation $Xcal\,Ycal$, $stdXcalYcal$



(a) Cetane concentration. (b) Sulfur concentration.

Fig. 4. Membership function for the correlation between *Xcal* and *Ycal* in oil samples

is the standard deviation from $Xcal$ and $Ycal$ correlation, $meanXcalYcal$ is the mean from $Xcal$ and $Ycal$ and $meanXcalYcalAbovePar$ is the mean from $Xcal$ and $Ycal$ correlation that are above par. Figure 4 we can see the geometry patterns of the membership functions for the second variable.



(a) Cetane concentration. (b) Sulfur concentration.

Fig. 5. Membership functions for the RMSEP in oil samples

The discourse universe of the third input variable validation error (RMSEP) is defined as [0.3 2.5] to cetane and as [0.004 0.04] to sulfur. We also, specified three membership functions for this variable : *Low*, *Medium* and *High*. Figure 5 presents the function geometric pattern for the concentrations of the oil samples. We could not define a empirically process for the error variable.

The output variable is the solution quality defined in the universe of discourse in the interval 0 to 10. This variable is independent of the component been analyzed by the system. Figure 6 shows the geometric patterns of the membership function specified *Poor*, *Medium* and *Good*.

The fuzzy inference system developed is composed by 27 rules of generalized *Modus Ponens* generalizado and they have the follow format:

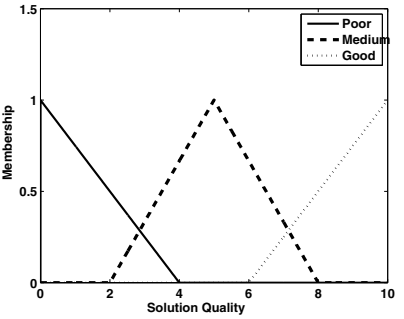


Fig. 6. Membership functions for the output variable solution quality

If (Correlation Xcal is Low) and (Correlation Xcal Ycal is Low) and (RMSEP is High) Then (Solution Quality is Poor)

the total set of rules can be seen in Table 1.

Table 1. Inference Rules Set

RMSEP is *Low*			
	Correlation Xcal Ycal is *Low*	Correlation Xcal Ycal is *Medium*	Correlation Xcal Ycal is *High*
Correlation Xcal is *Low*	*Medium*	*Good*	*Good*
Correlation Xcal is *Medium*	*Medium*	*Good*	*Good*
Correlation Xcal is *High*	*Poor*	*Poor*	*Medium*
RMSEP is *Medium*			
	Correlation Xcal Ycal is *Low*	Correlation Xcal Ycal is *Medium*	Correlation Xcal Ycal is *High*
Correlation Xcal is *Low*	*Medium*	*Good*	*Good*
Correlation Xcal is *Medium*	*Medium*	*Medium*	*Good*
Correlation Xcal is *High*	*Poor*	*Poor*	*Poor*
RMSEP is *High*			
	Correlation Xcal Ycal is *Low*	Correlation Xcal Ycal is *Medium*	Correlation Xcal Ycal is *High*
Correlation Xcal is *Low*	*Poor*	*Medium*	*Medium*
Correlation Xcal is *Medium*	*Poor*	*Medium*	*Poor*
Correlation Xcal is *High*	*Poor*	*Poor*	*Poor*

In order to preserve the diversity in the population we used the Crowding Distance Method proposed by Deb [2] with some changes to apply in the results of the fuzzy system. Infinity distance is set for individuals with the highest fuzzy value and with the best value of each fitness. The other individual was sorted and the Crowding distance was calculated by the difference between the solutions quality. The individuals with the biggest values of Crowding distance were selected.

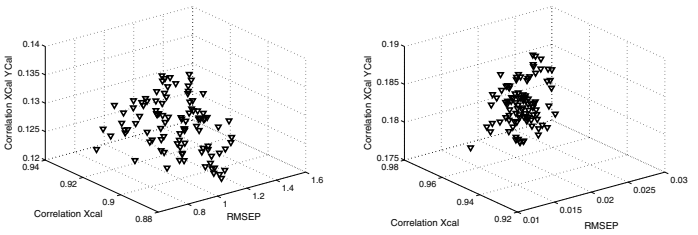
3 Results

The proposed Hybrid Genetic-Fuzzy algorithm for variable selection was used to determine the concentrations of cetane and sulfur in oil samples. The data set was obtained in the Campinas city from gas stations. The data set is composed

by 193 cetane samples, 126 sulphur samples and 4669 spectroscopy variables for both. The spectra (λ) were acquired in the range 750-2500nm wavelength.

The data set for cetane was divided as follow: 120 samples to make up the regression model, 30 samples to validate the model and 43 samples to calculate the prediction error in order to validate the obtained solutions. For sulfur the follow division was used: 88 samples to make up the regression model, 19 samples to validate de model and 19 samples to calculate the prediction error and validate the obtained solutions.

The tests were performed using the proposed hybrid multi-objective genetic algorithm and a standard MOGA that uses the non-dominance selection method. Both algorithms performed 100 generations with a population of 100 individuals. In each generation both algorithm selected the 20 best individuals to execute the recombination process. In this process the selected individuals recombine with the other individuals in the population in order to obtain a new population.



(a) Cetane concentrations. (b) Sulfur concentrations.

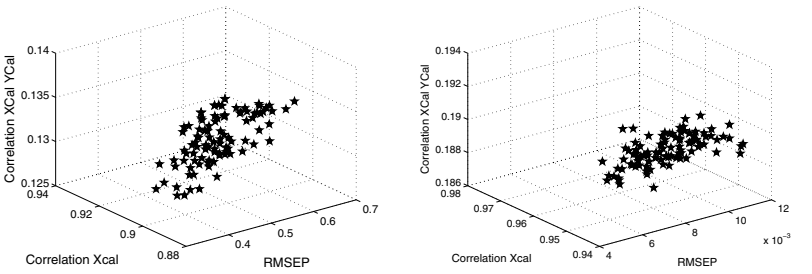
Fig. 7. Individuals obtained in the last generation of the proposed algorithm with Crowding Distance

Figures 7 and 8 show the final population of the proposed algorithm and the standard MOGA with their fitness values. In both results there was a good diversity in the population, but in the standard MOGA we do not have a good weighing among the different objectives.

Tables 2 and 3 show that despite the individuals from MOGA in comparison with the individuals from the proposed algorithm, there are better fitness values for the particular objectives in some solutions, the individuals from the hybrid algorithm have better weighting in the integrated evolution of the three objectives.

We can observe in Tables 2 and 3 that, despite the individuals from MOGA, in comparison with the individuals from the proposed algorithm, have better fitness values for the particular objectives in some solutions, the individuals from the hybrid algorithm have a better weighting in the integrated evolution of the three objectives.

Another important result about the individuals from the hybrid algorithm can be seen on Tables 4 and 5. These tables show the prediction error of the best five individuals to cetane and sulfur concentrations. The individuals from MOGA have a bigger prediction error than the individuals from the proposed algorithm.



(a) Cetane concentrations. (b) Sulfur concentrations.

Fig. 8. Individuals obtained in the last generation of MOGA

Table 2. Fitness values of the five best individuals for cetane concentrations

MOGA					
	Individual				
	1	2	3	4	5
Fitness 1$^{\diamond}$	0.9135	0.8955	0.9053	0.8914	0.8914
Fitness 2†	0.1344	0.1296	0.1335	0.1287	0.1287
Fitness 3‡	0.5626	0.3084	0.6842	0.3667	0.3667
Hybrid Genetic-Fuzzy Algorithm					
	Individual				
	1	2	3	4	5
Fitness 1$^{\diamond}$	0.9212	0.8851	0.9355	0.8900	0.8880
Fitness 2†	0.1240	0.1217	0.1304	0.1235	0.1228
Fitness 3‡	0.7075	1.0937	1.3332	1.0712	1.1422
Defuzzification	3.9505	4.9263	1.4965	4.7669	4.8340

Table 3. Fitness values of the five best individuals for sulfur concentrations

MOGA					
	Individual				
	1	2	3	4	5
Fitness 1$^{\diamond}$	0.9414	0.9449	0.9549	0.9494	0.9421
Fitness 2†	0.1891	0.1879	0.1905	0.1909	0.1895
Fitness 3‡	0.0108	0.0073	0.0074	0.0101	0.0110
Hybrid Genetic-Fuzzy Algorithm					
	Individual				
	1	2	3	4	5
Fitness 1$^{\diamond}$	0.9712	0.9378	0.9497	0.9379	0.9502
Fitness 2†	0.1859	0.1808	0.1798	0.1809	0.1811
Fitness 3‡	0.0252	0.0157	0.0112	0.0154	0.0185
Defuzzification	2.8192	6.3287	5.4369	6.3776	5.3174

$^{\diamond}$ Correlation between the variables in Xcal
† Correlation between the variables of Xcal Ycal
‡ RMSEP

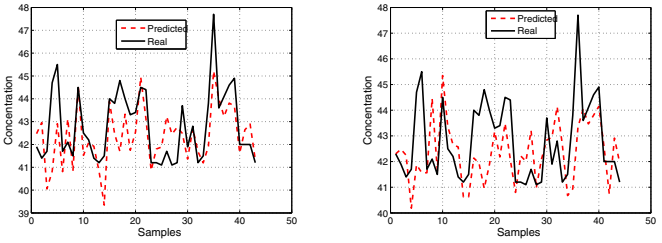
Figure 9 shows the prediction for the cetane concentrations to the 43 samples of oil. These samples did not participate on the evolution process and the real values of the samples were obtained by the laboratory process. Figure 9(a) shows

Table 4. RMSEP for cetane concentrations for the five individuals

	RMSEP	
	MOGA	Hybrid Algorithm
Individual 1	1.6216	1.3459
Individual 2	1.6810	1.4324
Individual 3	1.6843	1.4515
Individual 4	1.6108	1.3316
Individual 5	1.6108	1.4759

Table 5. RMSEP for sulfur concentrations for the five individuals

	RMSEP	
	MOGA	Hybrid Algorithm
Individual 1	0.0373	0.0372
Individual 2	0.0358	0.0333
Individual 3	0.0389	0.0348
Individual 4	0.0385	0.0332
Individual 5	0.0377	0.0344



(a) Prediction obtained with best solution of the Hybrid Genetic-Fuzzy algorithm.

(b) Prediction obtained with the solution with better RMSEP from MOGA.

Fig. 9. Prediction of cetane concentrations with multiple linear regression and variable selection

the prediction to the individual with highest fuzzy classification and Figure shows the prediction for the individual from MOGA with lowest RMSEP. These results validate the information from Table 4 that the solutions from MOGA are worse than that from the proposed algorithm for process prediction.

4 Conclusions

This paper proposed a hybrid genetic-fuzzy multi-objective algorithm for the problem of variable selection in spectroscopy. Firstly the problem of variable selection in multiple linear regression in data of multivariate nature was presented. Begging with the instrumental data obtained from spectrometer, this work proposed a substitution of non-domination selection method by the fuzzy evaluation

system. The results show that the proposed methodology obtained better results than the algorithm based on non-domination. Future works include the use of fourth objective, the number of variables in the model for reducing the variable set length.

References

1. Augugliaro, A., Dusonchet, L., Riva Sanseverino, E.: Multiobjective service restoration in distribution networks using an evolutionary approach and fuzzy sets. Electrical Power and Energy Systems 22(2), 103–110 (2000)
2. Deb, K.: Multi-objective optimization using evolutionary algorithms. John Wiley and Sons, Ltd. (2001)
3. Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Company, Inc., Reading (1989)
4. Guy, R.H., Hostynek, J.J., Hinz, R.S., Lorence, C.R.: Metals and the skin. Marcel Dekker Incorporated (1999)
5. Hsiao, Y.-T., Chien, C.-Y.: Enhancement of restoration service in distribution systems using a combination fuzzy-ga method. IEEE Transactions on Power Systems 15(4), 1394–1400 (2000)
6. Naes, T., Mevik, B.H.: Understanding the collinearity problem in regression and discriminant analysis. Journal of Chemometrics 15(4), 413–426 (2001)
7. Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms. In: Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms, pp. 93–100. Lawrence Erlbaum (1985)
8. Skoog, D.A.: Princpios de anlise instrumental. Bookman (2002)
9. Srinivas, N., Deb, K.: Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation 2(3), 221–248 (1994)
10. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)

Multiple Choice Strategy – A Novel Approach for Particle Swarm Optimization – Preliminary Study

Michal Pluhacek*, Roman Senkerik, and Ivan Zelinka

Tomas Bata University in Zlin, Faculty of Applied Informatics,
T.G. Masaryka 5555, 760 01 Zlin, Czech Republic
{pluhacek,senkerik,zelinka}@fai.utb.cz

Abstract. In this paper, a new promising strategy for PSO algorithm is proposed and described. This strategy presents alternative way of assigning new velocity to each particle in population. This new multiple choice particle swarm optimization (MC-PSO) algorithm is tested on four different test functions and the promising results of this alternative strategy are compared with the classic method.

1 Introduction

PSO algorithm is one of evolutionary algorithms, which are a class of soft computing methods inspired by nature. In past years and decades, various evolutionary algorithms were designed, modified and used with great results in many areas of optimization [1-8].

This preliminary study investigates on new strategy that is based on heuristic selection from multiple choices of particle behave. The primary aim of this strategy is to improve the performance of PSO algorithm especially when dealing with higher dimension problems. The theory behind this strategy is based on multiple observations of PSO algorithm behaviour under various conditions.

2 Particle Swarm Optimization Algorithm

PSO (Particle swarm optimization) algorithm is based on the natural behavior of birds and fishes and was firstly introduced by R. Eberhart and J. Kennedy

* This work was supported by Grant Agency of the Czech Republic - GACR P103/13/08195S, by the Development of human resources in research and development of latest soft computing methods and their application in practice project, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, cofinanced by ESF and state budget of the Czech Republic, European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089 and by Internal Grant Agency of Tomas Bata University under the project No. IGA/FAI/2013/012.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 36–45, 2013.

in 1995 [1-3]. As an alternative to genetic algorithms [4] and differential evolution [5], PSO proved itself to be able to find satisfactory solutions for many optimization problems [1-2, 6, 8]. Term "swarm intelligence" [2, 3] refers to the capability of particle swarms to exhibit surprising intelligent behavior assuming that some form of communication (even very primitive) can occur among the swarm particles (individuals).

Basic PSO algorithm disadvantage was the poor local search capability. For this reason, several modifications of PSO were introduced to handle with this problem. Main principles of PSO algorithm and its modifications are well described in [1-3, 6, and 8].

Within this research, PSO strategy with linear decreasing inertia weight [6] was used as a base for the new algorithm and also as an alternative algorithm to compare the results. The selection of inertia weight modification of PSO was based on numerous previous experiments. Default values of all PSO parameters were chosen according to the recommendations given in [2, 3]. Inertia weight is designed to influence the velocity of each particle differently over the time [6, 8]. In the beginning of the optimization process, the influence of inertia weight factor w is minimal. As the optimization continues, the value of w is decreasing, thus the velocity of each particle is decreasing, since w is always the number < 1 and it multiplies previous velocity of particle in the process of new velocity value calculation. In each generation, a new location of a particle is calculated based on its previous location and velocity (or "velocity vector"). Linear decreasing inertia weight PSO strategy has two control parameters w_{start} and w_{end}. New w for each generation is then given by Eq. 1, where i stand for current generation number and n for total number of generations.

$$w = w_{start} - \frac{((w_{start} - w_{end})\, i)}{n} \tag{1}$$

Inertia weight value is than applied in the main formula of PSO algorithm (Eq. 2) which determines a new "velocity", thus the position of each particle in the next generation (or migration cycle).

$$v(t+1) = w \cdot v(t) + c_1 \cdot Rand \cdot (pBest - x(t)) + c_2 \cdot Rand \cdot (gBest - x(t)) \tag{2}$$

Where:
$v(t+1)$ – New velocity of particle.
$v(t)$ – Current velocity of particle.
c_1,c_2 – Priority factors.
pBest – Best solution found by particle.
gBest – Best solution found in population.
$x(t)$ – Current position of particle.
Rand – Random number, from the interval <0,1>.

New position of a particle is then given by Eq. 3, where x(t+1) represents the new position:

$$x(t+1) = x(t) + v(t+1) \tag{3}$$

3 Multiple Choice Particle Swarm Optimization Algorithm (MC-PSO)

New strategy proposed in this study alters the original way (Eq. 2) of calculating the particle velocity in the next generation. At first, three numbers b_1, b_2 and b_3 are defined at the start of algorithm. These numbers represent limit values for different rules, so they should follow the pattern: $b_1 < b_2 < b_3$. In this study following values were used: $b_1 = 0.2$, $b_2 = 0.4$, $b_3 = 0.7$. Afterwards during the calculation of new velocity of each particle a random number r is generated from the interval <0, 1>. Finally the new velocity is calculated following these four rules:

If r $\leq b_1$ the new velocity of particle is given by Eq. 4:

$$v(t+1) = 0 \tag{4}$$

If $b_1 <$ r $\leq b_2$ the new velocity of particle is given by Eq. 5:

$$v(t+1) = w \cdot v(t) + c \cdot Rand \cdot (x_r(t) - x(t)) \tag{5}$$

Where $x_r(t)$ is the position of randomly chosen particle.

If $b_2 <$ r $\leq b_3$ the new velocity of particle is given by Eq. 6:

$$v(t+1) = w \cdot v(t) + c \cdot Rand \cdot (pBest - x(t)) \tag{6}$$

If $b_3 <$ r the new velocity of particle is given by Eq. 7:

$$v(t+1) = w \cdot v(t) + c \cdot Rand \cdot (gBest - x(t)) \tag{7}$$

The priority factors c_1 and c_2 from original equation (Eq. 2) are replaced within this novel aproach with c. In this novel strategy c defines not the priority (which is given by b_1, b_2 and b_3 setting) but the overstep value. In other words how far past the target (pBest, gBest or random particle) can the active particle go. Within this work, c was set to 2. Pseudocode of proposed algorithm is depicted on Figure 1.

4 Test Functions

Following set of four common test functions was used in this study.
The First De Jong‘s function is given by Eq. 8.

$$f(x) = \sum_{i=1}^{\dim} x_i^2 \tag{8}$$

Function minimum:
Position for E_n: $(x_1,x_2 \ldots x_n) = E_n$: $(x_1,x_2 \ldots x_n) = (0,0,\ldots, 0)$
Value for E_n: $y = 0$

1. Initialize population
2. Evaluate and assign *pBest* and *gBest*
3 . REPEAT UNTIL stopping condition met:
4. Calculate inertia weight constant *w*
5. FOR each individual:
6. *r* = RANDOM[0,1]
7. IF *($r \le b_1$) v(t+1)* = 0
8. IF *($b_1 < r \le b_2$) v(t+1)* = *w* • *v(t)* + *c* • RANDOM[0,1] • *($x_r(t)$ - x(t))*
9. IF *($b_2 < r \le b_3$) v(t+1)* = *w* • *v(t)* + *c* • RANDOM[0,1] • *(pBest- x(t))*
10. IF *($b_3 < r$) v(t+1)* = *w* • *v(t)* + *c* • RANDOM[0,1] • *(gBest - x(t))*
11. Evaluate and assign pBest and gBest
12. End

Fig. 1. Multiple Choice PSO - pseudocode

The Second De Jong‘s function is given by Eq. 9.

$$f(x) = \sum_{i=1}^{\dim -1} 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2 \tag{9}$$

Function minimum:
Position for E_n: $(x_1,x_2 \ldots x_n) = (1,1,\ldots,1)$
Value for E_n: $y = 0$

Rastrigin‘s function is given by Eq. 10.

$$f(x) = 10\dim + \sum_{i=1}^{\dim} x_i^2 - 10\cos(2\pi x_i) \tag{10}$$

Function minimum:
Position for E_n: $(x_1,x_2 \ldots x_n) = (0,0,\ldots,0)$
Value for E_n: $y = 0$

Schwefel‘s function is given by Eq. 11

$$f(x) = \sum_{i=1}^{\dim} -x_i \sin(\sqrt{|x|}) \tag{11}$$

Function minimum:
Position for E_n: $(x_1,x_2 \ldots x_n) = (420.969, 420.969,\ldots, 420.969)$
Value for E_n: $y = -418.983 \cdot \text{dimension}$

5 Experiment Setup

Within all performance testing two PSO strategies were used. The first one was the classic PSO with linear decreasing inertia weight (as described in section 2) noted PSO Weight. The second was the new multiple choice strategy described in section 3 (noted MC-PSO).

The above described test functions (section 6) were used for both strategies of PSO algorithm. For each strategy, 100 separate runs were performed and statistically analysed.

Control parameters were set up based on the previous numerous experiments and literature [1-3, 6, 8] as follows:

Population size: 50
Generations: 500
w_{start}: 0.9
w_{end}: 0.4
Dimension: 40, 100, 1000

6 Results

Following tables (Table 1-12) and figures (Fig. 2-4) show the results of the performance testing. Best mean CF values and best overall results in tables are highlighted. Brief analysis of results follows afterwards. Results for the 1^{st} De Jong's function are shown in Tables 1-3.

Results for the 2^{nd} De Jong's function are shown in Tables 4-6. Furthermore the history of gBest value was tracked and can be seen on Figures 2-4. Results for the Rastrigins function are shown in Tables 7-9. Results for the Schwefel's function are shown in Tables 10-12.

Table 1. Results for the 1^{st} De Jong's function (dim = 40)

Dim: 40	PSO Weight	MC-PSO
Mean CF Value:	1.96619	**0.716154**
Std. Dev.:	0.76231	0.280076
CF Value Median:	1.85412	0.668139
Max. CF Value:	4.99345	1.78528
Min. CF Value:	0.611592	**0.18597**

7 Brief Analysis of Results

From the results presented in section 6, it follows that the multiple choice strategy seems to have positive impact on the performance of PSO algorithm in

Table 2. Results for the 1^{st} De Jong's function (dim = 100)

Dim: 100	PSO Weight	MC-PSO
Mean CF Value:	22.867	**8.09569**
Std. Dev.:	4.46981	1.77228
CF Value Median:	22.7022	7.91636
Max. CF Value:	35.9867	13.4529
Min. CF Value:	14.7426	**4.8686**

Table 3. Results for the 1^{st} De Jong's function (dim = 1000)

Dim: 1000	PSO Weight	MC-PSO
Mean CF Value:	740.847	**201.926**
Std. Dev.:	71.2338	25.1704
CF Value Median:	736.41	199.824
Max. CF Value:	913.31	285.152
Min. CF Value:	600.914	**150.213**

Table 4. Results for the 2^{nd} De Jong's function (dim = 40)

Dim: 40	PSO Weight	MC-PSO
Mean CF Value:	363.102	**191.171**
Std. Dev.:	117.776	56.4784
CF Value Median:	341.793	185.947
Max. CF Value:	907.675	338.626
Min. CF Value:	154.146	**67.4299**

Table 5. Results for the 2^{nd} De Jong's function (dim = 100)

Dim: 100	PSO Weight	MC-PSO
Mean CF Value:	4746.66	**1148.44**
Std. Dev.:	1320.9	231.151
CF Value Median:	4648.1	1116.51
Max. CF Value:	8261.94	1703.05
Min. CF Value:	1861.05	**685.734**

Table 6. Results for the 2^{nd} De Jong's function (dim = 1000)

Dim: 1000	PSO Weight	MC-PSO
Mean CF Value:	310896	**32152.9**
Std. Dev.:	57751.6	5561.2
CF Value Median:	306360	31854.2
Max. CF Value:	481518	47096.9
Min. CF Value:	183916	**20515.8**

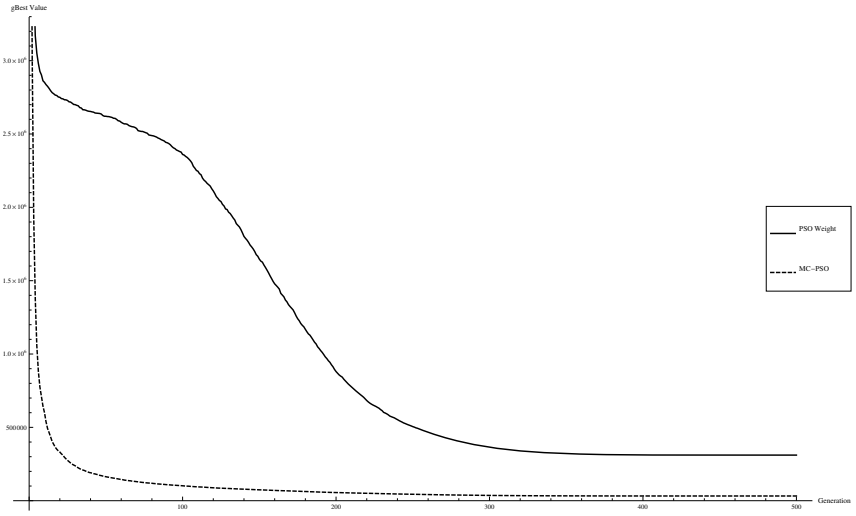


Fig. 2. History of mean best CF value for 100 runs - comparison

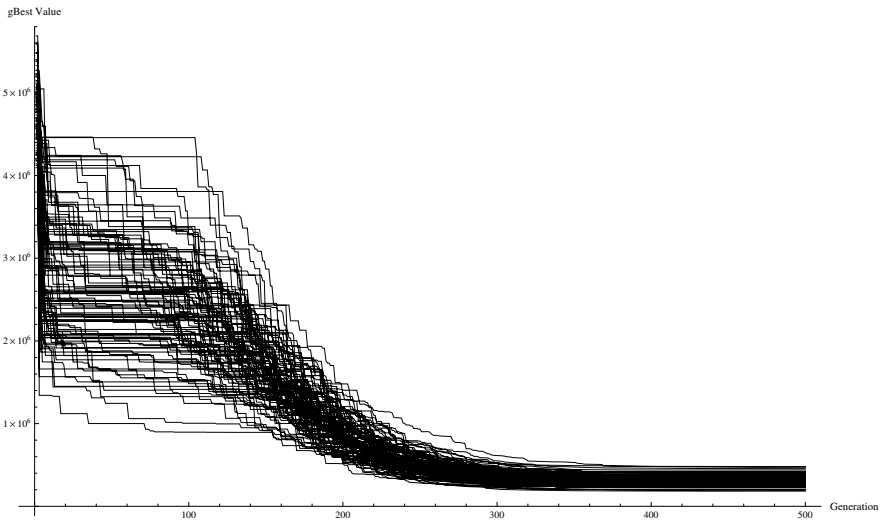


Fig. 3. History of mean best CF value for 100 runs – PSO Weight

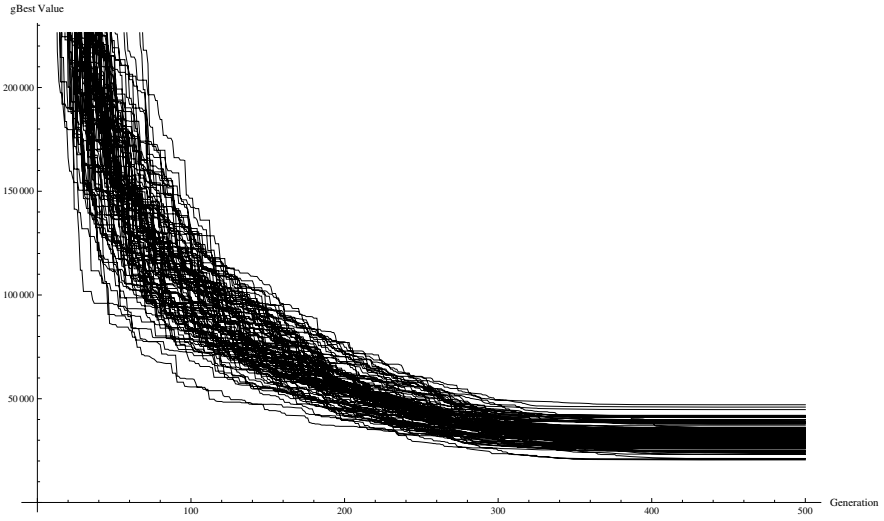


Fig. 4. History of mean best CF value for 100 runs – MC-PSO

Table 7. Results for the Rastrigin's function (dim = 40)

Dim: 40	PSO Weight	MC-PSO
Mean CF Value:	114.429	**113.944**
Std. Dev.:	22.0365	24.1512
CF Value Median:	115.635	111.306
Max. CF Value:	171.802	199.115
Min. CF Value:	73.1292	**70.6913**

Table 8. Results for the Rastrigin's function (dim = 100)

Dim: 100	PSO Weight	MC-PSO
Mean CF Value:	551.721	**513.69**
Std. Dev.:	41.1921	46.1446
CF Value Median:	558.379	514.081
Max. CF Value:	633.897	615.184
Min. CF Value:	421.706	**377.552**

Table 9. Results for the Rastrigin's function (dim = 1000)

Dim: 1000	PSO Weight	MC-PSO
Mean CF Value:	9440.8	**8955**
Std. Dev.:	189.346	215.92
CF Value Median:	9440.98	8964.37
Max. CF Value:	10036.5	9584.79
Min. CF Value:	9016.3	**8428.7**

Table 10. Results for the Schwefel's function (dim = 40)

Dim: 40	PSO Weight	MC-PSO
Mean CF Value:	-5989.72	**-7769.11**
Std. Dev.:	652.058	674.048
CF Value Median:	-5963.77	-7816.93
Max. CF Value:	-4776.98	-6336.23
Min. CF Value:	-7584.36	**-9536.35**

Table 11. Results for the Schwefel's function (dim = 100)

Dim: 100	PSO Weight	MC-PSO
Mean CF Value:	-9708.52	**-13399.9**
Std. Dev.:	1099.22	1073.53
CF Value Median:	-9779.33	-13368.2
Max. CF Value:	-7151.08	-11035.7
Min. CF Value:	-13164.8	**-16026.1**

Table 12. Results for the Schwefel's function (dim = 1000)

Dim: 1000	PSO Weight	MC-PSO
Mean CF Value:	-30697.7	**-45825**
Std. Dev.:	3417.37	3823.02
CF Value Median:	-30624.7	-45618.4
Max. CF Value:	-23970	-37081
Min. CF Value:	-39961.4	**-54014.6**

higher dimensions. In every test, the MC-PSO managed to find better solution than the classis PSO Weight strategy. Furthermore, Figures 2-4 lend weight to the argument, that the multiple choice strategy appears to significantly change the overall behavior of PSO algorithm especially in terms of convergence speed.

8 Conclusion

In this research a new multiple choice strategy for PSO algorithm was proposed. This strategy allows each particle to make one of four predefined possible actions. Each particle is capable of stopping its movement or moving in the direction of randomly selected particle, pBest or gBest. This approach was tested on four standard benchmark functions with very promising results. These results were compared with canonical PSO strategy with linear decreasing inertia weight. This canonical strategy served as a basis for the new multiple choice strategy. The proposed new strategy managed to find better solutions in all tests. Future testing is needed to bring more evidence to support this idea.

References

1. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. IV, pp. 1942–1948 (1995)
2. Eberhart, R., Kennedy, J.: Swarm Intelligence. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann (2001)
3. Dorigo, M.: Ant Colony Optimization and Swarm Intelligence. Springer (2006)
4. Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning, p. 41. Addison Wesley (1989) ISBN 0201157675
5. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)
6. Shi, Y.H., Eberhart, R.C.: A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation, Anchorage Alaska, pp. 69–73 (1998)
7. Zelinka, I.: SOMA—self organizing migrating algorithm. In: Babu, B.V., Onwubolu, G. (eds.) New Optimization Techniques in Engineering, vol. 33, ch. 7. Springer (2004) ISBN: 3-540-20167X
8. Nickabadi, A., Ebadzadeh, M.M., Safabakhsh, R.: A novel particle swarm optimization algorithm with adaptive inertia weight. Applied Soft Computing 11(4), 3658–3670 (2011) ISSN 1568-4946

Method of Handling Constraints in Differential Evolution Using Fletcher's Filter

Wojciech Rafajłowicz

Wojciech Rafajłowicz is with the Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50 370 Wrocław, Poland
wojciech.rafajlowicz@pwr.wroc.pl

Abstract. In this paper we propose a new approach to solving constrained nonlinear optimization problems using differential evolution. A method of handling the constraints is based on the works of Fletcher and his co-workers. The filter is a general methodology of handling constraints and differential evolution is a very efficient method for unconstrained optimization problems. Thus, one can expect a high efficiency of using them together. The filter is based on the idea of multi-criteria optimization. Numerical results for typical benchmark problems are provided. The efficiency of the proposed method ocureed satisfactory.

1 Introduction

Constrained nonlinear optimization problems are encountered in many different scientific and engineering problems. Differential evolution is a method proposed by Kenneth Price and Reiner Stron in a technical report [12] in 1995. From that time the method becomes increasingly popular as one of the most efficient methods. This method provides better results then a simple evolution retaining its abilities to search for non-local solutions.

The filter was proposed by Fletcher and his co-workers in 2001, firstly for sequential linear programming and then for sequential quadratic programming. Today Filter SQP method is considered as a state of the art method in classical constrained optimization.

We are convinced that the methodology of applying the filter is much wider and can be useful in developing new algorithms for many optimization problems with constraints. Our aim in this paper is to propose a new approach that incorporates the filter into differential evolution algorithms.

The idea of a filter was recently used in random search meta-heuristic methods for the first time in evolutionary method [16] and then in [17] as a method of holding constraints.

Many methods of handling constraints in differential evolution were proposed. Some of them are simple penalty function or merit function as evaluated in [9]. In [11] a complicated set of comparison criterions has been proposed that are close in spirit to the filter idea. Most further work like [7] concentrate on initial population and special cross-over methods. Others, like [8] propose to get more

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 46–55, 2013.

vectors for reproduction from the population. It is sometimes proposed (in the same paper) to run algorithms simultaneously in combination. Results from this proposition were quite satisfactory.

2 Problem Statement

The classical constrained nonlinear programming problem is as follows

$$\text{minimize } f(\mathbf{x}) \text{ subject to} \tag{1}$$

$$\begin{aligned} g_1(\mathbf{x}) &\leq 0 \\ &\vdots \\ g_i(\mathbf{x}) &\leq 0 \\ &\vdots \\ g_m(\mathbf{x}) &\leq 0 \end{aligned} \tag{2}$$

where $x \in R^n$, $i = 1, \ldots, m$.

Such problems are common in many engineering tasks like optimal control and plant design. Many other problems, including some image processing tasks, can be formulated in an optimization problem form. In extreme cases of optimal control or shape optimization they can have a vector x of a couple of hundreds of elements and thousands of constraints.

Typically smaller benchmark problems like `g01` – `g10` proposed by Schittkowski are used to test methods and their implementation.

Without losing the generality, we confine ourselves to the inequality constraints, because equality constraints can be taken into account by adding two inequality constraints.

Define a penalty function, denoted further by h,

$$h(g(\mathbf{x})) = \sum_{j=1}^{m} \max\left(0,\, g_j(\mathbf{x})\right). \tag{3}$$

Note that for $\mathbf{x} \notin \mathcal{C}$ the penalty $h(g(\mathbf{x})) > 0$ is a measure of the constraints violation. We also have $h(g(\mathbf{x})) = 0$ if and only if $\mathbf{x} \in \mathcal{C}$. Observe that $h(g(\mathbf{x}))$ is not differentiable.

3 Filter

The filter is a data structure that aggregates information about method's previous temporary solutions. It can be used to assess the quality of current solution. The notion of a filter was introduced by Fletcher and his co-workers (see [2], [3]) as a tool for solving constrained optimization problems by generating sequences that are solutions of quadratic or linear approximations to $f(x)$ and linear approximations to $g(x)$.

A filter presented in this paper follows the original Fletcher idea (not the new one, but slightly different version proposed in [4]) with minor modifications.

Definition 1. *In k-th generation a filter $\mathcal{F}_k$ is a list of pairs $(h_k,\, f_k)$, which were generated according to the rules described below, where for given $\mathbf{x}_k$ we denote by $(h_k,\, f_k)$ a pair of the form: $(h(\mathbf{c}(\mathbf{x}_k)),\, f(\mathbf{x}_k))$. The list of $\mathbf{x}_k$'s, which correspond to $(h_k,\, f_k) \in \mathcal{F}_k$ is also attached to the filter, but it is not displayed.*

In the above mentioned papers the authors avoid keeping x_k's in the filter. In our case it is necessary to store x_k's together with the corresponding $(h_k,\, f_k)$'s. This is not as memory consuming as one can expect, because we do not allow the filter content to grow above a certain level.

We say that a pair $(h_k,\, f_k)$ dominates $(h_l,\, f_l)$ if and only if

$$f_k \le f_l \quad \text{AND} \quad h_k \le h_l \tag{4}$$

and at least one of these inequalities is strict.

We shall need a somewhat more demanding notion of a dominance between such pairs as discussed later. At k-th generation the following rules govern the behavior of filter $\mathcal{F}_k$.

Rule 1) Filter $\mathcal{F}_k$ contains only pairs $(h_l,\, f_l)$, which were generated up to k-th generation and the corresponding $\mathbf{x}_l$'s. No pair in $\mathcal{F}_k$ dominates any other in the sense (4).

Rule 2) A new pair $(h_j,\, f_j)$ is allowed to be included to $\mathcal{F}_k$, if it is not dominated by any point already contained in $\mathcal{F}_k$.

Rule 3) If for a pair $(h_l,\, f_l)$ R2) holds, then it is acceptable for inclusion in the filter $\mathcal{F}_k$ and all entries dominated by this pair must be removed from $\mathcal{F}_k$ so as to ensure R1).

Typical content of the filter after succesful finishing optimization can be seen in fig. 4.

Remark 1. *Additionally we should consider additional barrier elements in the filter. The first of them $(f = -\infty, h = \infty)$ does not allow SE escape. In real world numbers we had used $(-1e99, 1e3)$. Not allowing NW escape is more complicated. We have to predict the upper bound for $f_{ub} = \max f(\mathbf{x})$*

Remark 2. *We should also note that the filter approaches constraints from outside of the feasible region. This means that a successful run should be defined as giving very small h. We prefer of course $h = 0$ but usually we are able to say what order of magnitude of violation is acceptable.*

4 Differential Evolution Approach

Differential evolution is a meta-heuristic method of optimization. In this method gradients are not directly computed but two random agents from the population are selected on a random basis. Then their difference is used with some coeficient (F). This method can be easily parallelized. It can be also modified to solve many types of problems. This method is called evolutionary however it is different then

a typical biologically inspired evolution presented for example in [5], [6] Here we propose a method for constrained optimization.

In a typical case differential evolution requires a population consisting of vectors from R^n which is the domain of goal function $f(x)$. In our case additional agents are stored in the filter.

In this approach the differentiated elements come from population and the modified element from the filter, so the method is as follows.

- Choose parameters: for differential evolution CR, F and for the filter: size of population and (possibly) maximal size of the filter. Initialize initial population and insert at least one element in the filter, possibly with small h.
- Until reaching a stop criterion, for each element in the population x

  Step 1 Choose at random an element from the filter and denote it by a. Choose two elements from the population b and c such that $b \neq c$. Choose a random number R from $1 \ldots n$ (the current working dimension for all vectors).

  Step 2 For each dimension $k = 1 \ldots n$
  1. Choose random r from $[0, 1]$
  2. If $r < CR$ or $k = R$ then $y_k = a_k + F \cdot (b_k - c_k)$
  3. else $y_k = x_k$

  Step 3 For resulting y calculate $f(y)$ and $h(g(y))$. If pair (f, h) is acceptable to the filter replace element x by y. Add triple (f, h, y) to the filter. Otherwise do not change the population

Remark 3. *We can clearly see similarities of these rules to those in [11]. The main difference is in storing elements with goal function–constraint violation trade-off in the filter. It will be clearer in numerical experiments.*

Remark 4. *The initial element of the filter has to be chosen. We propose to minimize the penalty $h(\mathbf{x})$ using any reasonable evolutionary algorithm for unconstrained minimization, expecting that we obtain points, which are in $\mathcal{C}$ or*

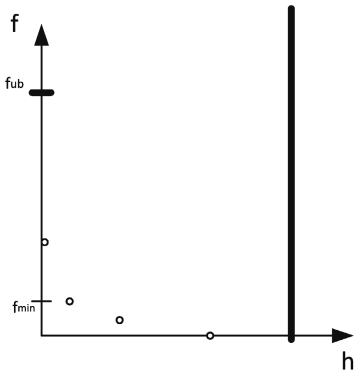


Fig. 1. Example of a filter content: dots indicate points included in the filter

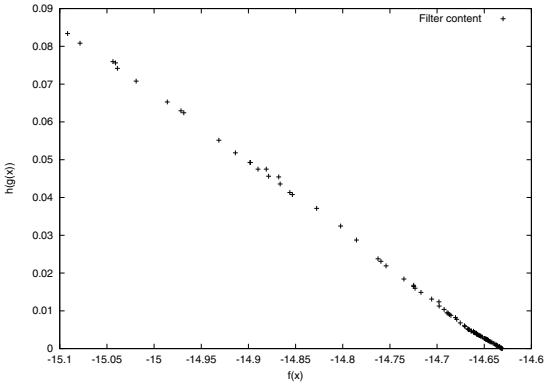


Fig. 2. Example of the filter contents after 1000 for g01 problem

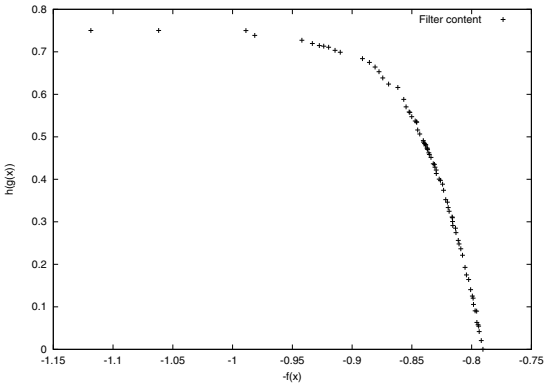


Fig. 3. Example of the filter contents after 1000 for g02 problem

close to it in the sense that their penalties are not larger than $\gamma > 0$. *In some of our simulations reported below we have used the* `Mathematica ver. 7` *function* `NMinimize` *with the option* `RandomSearch`.

Two additional problems occur in differential evolution. One of them is selecting the initial population. In our numerical experiments it occurred that it is enough to select a random population delimited by highest from box constraints of the problem.

Another one is selecting parameters CR and F for the algorithm. Usually $CR \in [0, 1]$ and $F \in [0, 2]$. Sometimes the easiest way is to sweep thr whole parameter space and choose the best solution.

We must also take measures not to use barrier elements present in the filter in calculation – we exclude them from selection.

We could also change the differential mutation method. The method used in this paper is the most basic one.

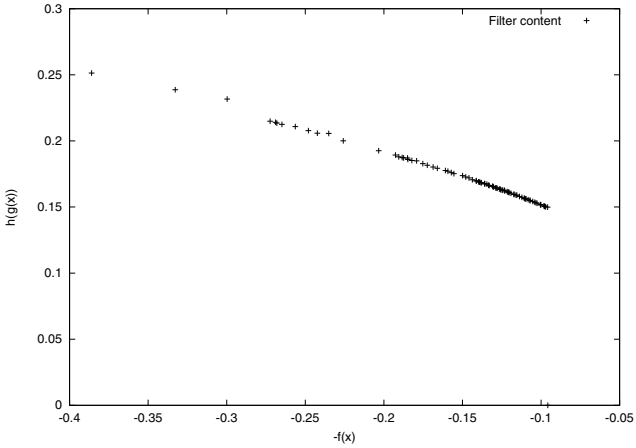


Fig. 4. Example of the filter contents

5 Numerical Results

Benchmark problems were originally proposed by Schittkowski and can be found in many papers such as [10]. Here we show the results for some of them.

The first of them is `g01`. This is a minimization problem with 13 variables.

$$f(\mathbf{x}) = 5\sum_{i=1}^{4}\mathbf{x}_i - 5\sum_{i=1}^{4}\mathbf{x}_i^2 - \sum_{i=5}^{13}\mathbf{x}_i \tag{5}$$

subject to

$$g_1(\mathbf{x}) = 2\mathbf{x}_1 + 2\mathbf{x}_2 + \mathbf{x}_{10} + \mathbf{x}_{11} - 10 \leq 0$$
$$g_2(\mathbf{x}) = 2\mathbf{x}_1 + 2\mathbf{x}_3 + \mathbf{x}_{10} + \mathbf{x}_{12} - 10 \leq 0$$
$$g_3(\mathbf{x}) = 2\mathbf{x}_2 + 2\mathbf{x}_3 + \mathbf{x}_{11} + \mathbf{x}_{12} - 10 \leq 0$$
$$g_4(\mathbf{x}) = -8\mathbf{x}_1 + \mathbf{x}_{10} \leq 0$$
$$g_5(\mathbf{x}) = -8\mathbf{x}_2 + \mathbf{x}_{11} \leq 0$$
$$g_6(\mathbf{x}) = -8\mathbf{x}_3 + \mathbf{x}_{12} \leq 0$$
$$g_7(\mathbf{x}) = -2\mathbf{x}_4 - \mathbf{x}_5 + \mathbf{x}_{10} \leq 0$$
$$g_8(\mathbf{x}) = -2\mathbf{x}_6 - \mathbf{x}_7 + \mathbf{x}_{11} \leq 0$$
$$g_9(\mathbf{x}) = -2\mathbf{x}_8 - \mathbf{x}_9 + \mathbf{x}_{12} \leq 0$$

additional bounds are $0 \leq \mathbf{x}_i \leq 1$ for $i = 1, \ldots, 9, 13$ and $0 \leq \mathbf{x}_i \leq 100$ for $i = 10, 11, 12$.

The method found a nearly optimal solution. The final filter content can be seen on fig. 2.

Parameters providing the best results were $CR = 0.5$, $F = 0.1$.

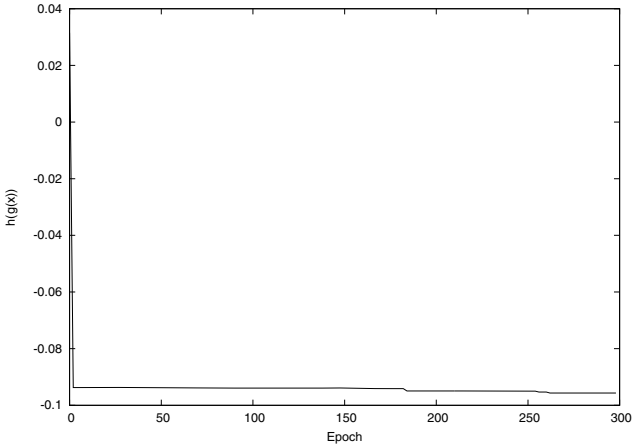


Fig. 5. Best values in subsequent epochs

Problem g02 is a maximization task. Our method is for minimization so we have to minimize $-f(\mathbf{x})$ where $f(\mathbf{x})$ is defined as follows

$$f(\mathbf{x}) = \left| \frac{\sum_{i=1}^{4} \cos^4(\mathbf{x}_i) - 2\Pi_{i=1}^{n} \cos^2(\mathbf{x}_i)}{\sqrt{\sum_{i=1}^{n} i\mathbf{x}_i^2}} \right| \tag{6}$$

subject to nonlinear constraints

$$g_1(\mathbf{x}) = 0.75 - \Pi_{i=1}^{n} \mathbf{x}_i \leq 0$$

$$g_2(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{x}_i - 7.5n \leq 0$$

where $n = 20$ and additionally each $\mathbf{x}_i$ has box constraints $0 \leq x_i \leq 10$.

The first entry in the filter was selected at random (from $[0, 10]$) and it was enough for the method to give acceptable $h = 0$ result for every CR and F. By sweeping parameter space we choose best $CR = 0.8$, $F = 0.1$. Calculation time was 40 seconds for 800 iterations and 120 agents in the population.

Statistical simulations were carried out and conclusions are drawn from 100 trial runs. Resulting empirical distribution of $f*$ (histogram) was shown on fig. 5.

Benchmark g08 is simpler because it has only two dimensions. On other hand it has many local minima, so the ability of evolutionary algorithms for such problems can be tested. Again it is a maximization problem so we have to minimize $-f(\mathbf{x})$

$$f(\mathbf{x}) = \frac{\sin^3(2\pi\mathbf{x}_1)\sin(2\pi\mathbf{x}_2)}{\mathbf{x}_1^3(\mathbf{x}_1 + \mathbf{x}_2)} \tag{7}$$

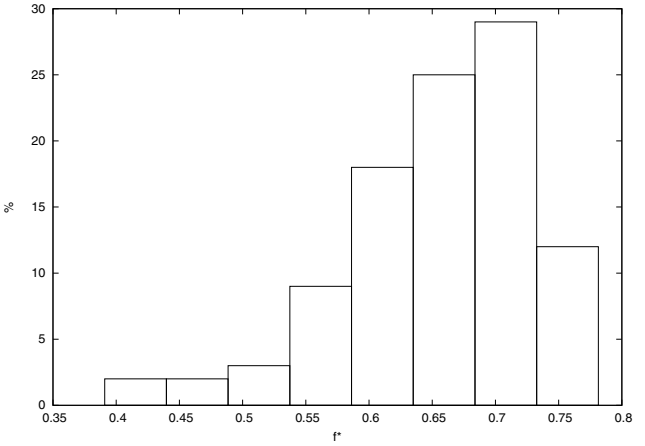


Fig. 6. Problem g02 results histogram (see explanations in the text)

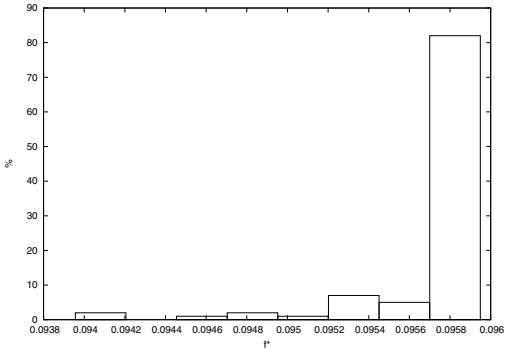


Fig. 7. Problem g08 results histogram (see explanations in the text)

subject to

$$g_1(\mathbf{x}) = \mathbf{x}_1^2 - \mathbf{x}_2 + 1 \leq 0$$

$$g_2(\mathbf{x}_) = 1 - \mathbf{x}_1 + (\mathbf{x}_2 - 4)^2 \leq 0$$

In fig. 8 we can see the contour plot of f with constraints. A small area between red lines is the feasible region. It is clear that this two-dimensional problem is not so easy due to many local extrema. On other hand the optimal solution is in the feasible region, so constraints are not active. In this situation the proposed method falls quickly into the feasible region. Then optimization becomes much simpler. It actually changes into a locally unconstrained problem.

Numerical results are quite interesting. When sweeping parameter space the differences between solutions are very small, about 10^{-4}.

The maximum (found as $-\min_x -f(\mathbf{x})$) has been found very quickly. Statistical simulations were carried out and conclusions are drawn from 100 trial runs. Resulting empirical distribution of $f*$ (histogram) was shown on fig. 7.

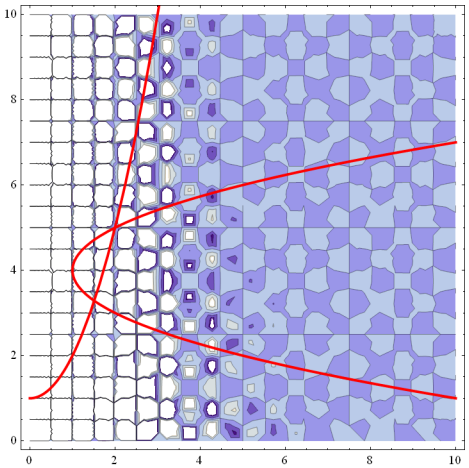


Fig. 8. Problem g08 contour plot with nonlinear constraints (in red)

6 Concluding Remarks

Differential evolution with a filter has turned out to be quite good metaheuristic method of nonlinear optimization with constraints. Typically for any random search method we can encounter phenomena known as the curse of dimensionality. Still, the method works fine for a 20-dimensional problem (`g02`). Further research should be carried out to find some rules for selection method parameters or testing methods proposed for unconstrained optimization.

The Fletcher filter in conjunction with sequential quadratic programing is able to solve nonlinear optimization problems arising in searching for optimal control signals for systems described by partial differential equations (see [15]). One may hope that also the approach proposed here can be sufficiently efficient in solving such problems, using the guidelines of applying stochastic search methods that have been proposed in [18], [19].

Differential evolution with filter can also be used in difficult learning problems, such as described in [1].

Calculations have been carried out in Wroclaw Centre for Networking and Supercomputing (http://www.wcss.wroc.pl).

References

1. Cpałka, K., Rutkowski, L.: Evolutionary learning of flexible neuro-fuzzy structures. In: Recent Advances in Control and Automation, Akademicka Oficyna Wydawnicza EXIT, pp. 398–407 (2008)
2. Fletcher, R., Leyffer, S., Toint, P.L.: On the global convergence of a filter-SQP algorithm. SIAM J. Optim. 13, 44–59 (2002)

3. Fletcher, R., Gould, N.I.M., Leyffer, S., Toint, P.L., Wachter, A.: Global convergence of trust-region SQP-filter algorithms for general nonlinear programming. SIAM J. Optimization 13, 635–659 (2002)
4. Roger Fletcher, A.: Sequential Linear Constraint Programming algorithm for NLP. SIAM Journal of Optimization 22(3), 772–794
5. Galar, R.: Handicapped Individua in Evolutionary Processes. Biol. Cybern. 53, 1–9 (1985)
6. Galar, R.: Evolutionary Search with Soft Selection. Biol. Cybern. 60, 357–364 (1989)
7. Gong, W., Cai, Z.: A Multiobjective Differential Evolution Algorithm for Constrained Optimization. In: 2008 IEEE Congress on Evolutionary Computation, CEC 2008 (2008)
8. Gordián-Rivera, L.-A., Mezura-Montes, E.: A Combination of Specialized Differential Evolution Variants for Constrained Optimization. In: Pavón, J., Duque-Méndez, N.D., Fuentes-Fernández, R. (eds.) IBERAMIA 2012. LNCS, vol. 7637, pp. 261–270. Springer, Heidelberg (2012)
9. de Melo, V., Grazieli, L., Costa, C.: Evaluating differential evolution with penalty function to solve constrained engineering problems. Expert Systems with Applications 39, 7860–7863 (2012)
10. Mezura-Montes, E., Coello, C.A.: A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems. IEEE Transactions on Evolutionary Computation 9(1), 1–17 (2005)
11. Mezura-Montes, E., Coello Coello, C.A., Tun-Morales, E.I.: Simple Feasibility Rules and Differential Evolution for Constrained Optimization. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS (LNAI), vol. 2972, pp. 707–716. Springer, Heidelberg (2004)
12. Storn, R., Price, K.: Differential evolution a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report (1995)
13. Storn, R., Price, K.: Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)
14. Price, K., Storn, R., Lampinen, J.: Differential Evolution A Practical Approach to Global Optimization. Springer, Heidelberg (2005)
15. Rafajłowicz, E., Styczeń, K., Rafajłowicz, W.: A modified filter SQP method as a tool for optimal control of nonlinear systems with spatio-temporal dynamics. International Journal of Applied Mathematics and Computer Science 22(2) (2012)
16. Rafajłowicz, E., Rafajłowicz, W.: Fletcher's Filter Methodology as a Soft Selector in Evolutionary Algorithms for Constrained Optimization. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) EC 2012 and SIDE 2012. LNCS, vol. 7269, pp. 333–341. Springer, Heidelberg (2012)
17. Rocha, A.M.A.C., Costa, M.F.P., Fernandes, E.M.G.P.: An Artificial Fish Swarm Filter-Based Method for Constrained Global Optimization. In: Murgante, B., Gervasi, O., Misra, S., Nedjah, N., Rocha, A.M.A.C., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2012, Part III. LNCS, vol. 7335, pp. 57–71. Springer, Heidelberg (2012)
18. Skowron, M., Styczeń, K.: Evolutionary search for globally optimal constrained stable cycles. Chemical Engineering Science 61(24), 7924–7932 (2006)
19. Skowron, M., Styczeń, K.: Evolutionary search for globally optimal stable multicycles in complex systems with inventory couplings. International Journal of Chemical Engineering (2009)

Chaos Driven Differential Evolution with Lozi Map in the Task of Chemical Reactor Optimization

Roman Senkerik[1,⋆], Donald Davendra[2], Ivan Zelinka[2], Michal Pluhacek[1], and Zuzana Kominkova Oplatkova[1]

[1] Tomas Bata University in Zlin, Faculty of Applied Informatics, Nam T.G. Masaryka 5555, 760 01 Zlin, Czech Republic
{senkerik,pluhacek,oplatkova}@fai.utb.cz
[2] Technical University of Ostrava, Faculty of Electrical Engineering and Computer Science, 17. listopadu 15,708 33 Ostrava-Poruba, Czech Republic
{donald.davendra,ivan.zelinka}@vsb.cz

Abstract. In this paper, Differential Evolution (DE) is used in the task of optimization of batch reactor geometry. The novality of the approach is that a discrete chaotic Lozi map is used as the chaotic pseudo random number generator to drive the mutation and crossover process in DE. The results obtained are compared with original reactor geometry and process parameters adjustment. The statistical analysis of the results given by chaos driven DE is compared with canonical DE strategy.

Keywords: Differential evolution, Chaos, Lozi map, Optimization, Reactor geometry.

1 Introduction

These days the methods based on soft computing such as neural networks, evolutionary algorithms, fuzzy logic, and genetic programming are known as powerful tool for almost any difficult and complex optimization problem. They are often applied in engineering design problems [1] or for the purpose of optimization of processing plants or technological devices [2] – [5]. Chemical industry produces a whole range of products through chemical reactions. Generally, it may be stated that key technological points are chemical reactors. Designing optimal reactor parameters including control constitutes is one of the most complex tasks in process engineering.

⋆ This work was supported by European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089, the project IT4Innovations Centre of Excellence No. CZ.1.05/1.1.00/02.0070, Grant Agency of the Czech Republic: GACR P103/13/08195S, and by the Development of human resources in research and development of latest soft computing methods and their application in practice project: No. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 56–66, 2013.

This paper is aimed at investigating the chaos driven DE. Although a number of DE variants have been recently developed, the focus of this paper is the embedding of chaotic systems in the form of chaos pseudo random number generator (CPRNG) for DE and its application to optimization of batch reactor geometry.

This research is a continuation of the previous successful initial application based experiment with chaos driven DE [6] and extends the initial research of chemical reactor optimization by means of chaos driven evolutionary algorithms [7] with the performance comparison of chaos driven DE against canonical DE strategy.

The chaotic systems of interest are discrete dissipative systems. The Lozi map was selected as the chaos pseudo random number generator for DE based on the successful results obtained with DE [6] or PSO algorithm [8].

Firstly, batch processes are explained. The next sections are focused on the description of the batch reactor, differential evolution, used chaotic systems and problem design. Results and conclusion follow afterwards.

2 Characteristics of the Batch Processes

The optimization of batch processes has attracted attention in recent years [9], [10]. Batch and semi-batch processes are of considerable importance in the fine chemicals industry. A wide variety of special chemicals, pharmaceutical products, and certain types of polymers are manufactured in batch operations. In batch operations, all the reactants are charged in a tank initially and processed according to a pre-determined course of action during which no material is added or removed. From a process systems point of view, the key feature that differentiates continuous processes from batch and semi-batch processes is that continuous processes have a steady state, whereas batch and semi-batch processes do not [11], [12].

Lot of modern control methods for chemical reactors were developed embracing the approaches such as iterative learning model predictive control [13], iterative learning dual-mode control [14] or adaptive exact linearization by means of sigma-point kalman filter [15]. Also the fuzzy approach is relatively often used[16]. Finally the methods of artificial intelligence are very frequently discussed and used. Many papers present successful utilization of either neural networks [17] – [19] or genetic (evolutionary) algorithms [20] – [22].

This paper presents the static optimization of the batch reactor by means of chaos driven differential evolution with chaotic Lozi map.

3 Description of the Reactor

This research uses a mathematical model of the reactor shown in Fig. 1. The reactor has two physical inputs: one for chemical substances **Chemical FK** (**Filter Cake**) with parameters temperature - T_{FK}, mass flow rate - $\dot{m}_{FK}$ and specific heat - c_{FK}; and one for cooling medium of temperature T_{VP}, mass flow

rate-$\dot{m}_V$ and specific heat - c_V. The reactor has one output: cooling medium flows through the jacket inner space of the reactor, with volume related to mass - m_{VR}, and flows out through the second output, with parameters mass flow rate m_V, temperature - T_V and specific heat - c_V.

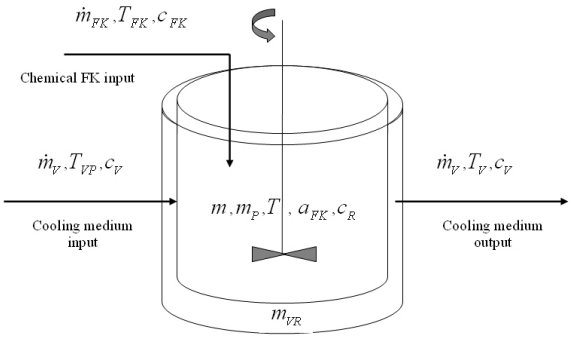


Fig. 1. Batch reactor

At the beginning of the process there is an initial batch inside the reactor with parameter mass-m_P. The chemical FK is then added to this initial batch, so the reaction mixture inside the reactor has total mass-m, temperature-T and specific heat-c_R, and also contains partially unreacted portions of chemical FK described by parameter concentration a_{FK}.

This "batch" technique partially allows controlling the temperature of reaction mixture by the controlled feeding of the input chemical FK. In general, this reaction is highly exothermal, thus the most important parameter is the temperature of the reaction mixture. **This temperature must not exceed 100°C** because of safety aspects and quality of the product. The original design of the reactor was based on standard chemical-technological methods and gives a proposal of reactor physical dimensions and parameters of chemical substances. These values are called within this paper **original parameters**.

The main objective of the optimization is to achieve the processing of large amount of chemical FK in a very short time.

Description of the reactor applies a system of four balance equations (1) and one equation (2) representing the term "k".

$$\dot{m}_{FK} = m'[t]$$

$$\dot{m}_{FK} = m\,[t]\; a'_{FK}[t] + k\, m[t]\, a_{FK}[t]$$

$$\dot{m}_{FK} c_{FK} T_{FK} + \Delta H_r k\, m\,[t]\, a_{FK}[t] = K\, S\,(T[t] - T_V[t]) + m[t]\, c_R\, T'[t]$$

$$\dot{m}_V c_V T_{VP} + K\, S\,(T[t] - T_V[t]) = \dot{m}_V c_V T_V[t] + m_{VR} c_V\, T'_V[t] \qquad (1)$$

$$k = A\, e^{-\frac{E}{R\,T[t]}} \qquad (2)$$

4 Differential Evolution

DE is a population-based optimization method that works on real-number-coded individuals [23]. For each individual $\boldsymbol{x}_{i,G}$ in the current generation G, DE generates a new trial individual $\boldsymbol{x}'_{i,G}$ by adding the weighted difference between two randomly selected individuals $\boldsymbol{x}_{r1,G}$ and $\boldsymbol{x}_{r2,G}$ to a randomly selected third individual $\boldsymbol{x}_{r3,G}$. The resulting individual $\boldsymbol{x}'_{i,G}$ is crossed-over with the original individual$\boldsymbol{x}_{i,G}$. The fitness of the resulting individual, referred to as a perturbed vector $\boldsymbol{u}_{i,G+1}$, is then compared with the fitness of $\boldsymbol{x}_{i,G}$. If the fitness of $\boldsymbol{u}_{i,G+1}$ is greater than the fitness of $\boldsymbol{x}_{i,G}$, then $\boldsymbol{x}_{i,G}$ is replaced with$\boldsymbol{u}_{i,G+1}$; otherwise, $\boldsymbol{x}_{i,G}$ remains in the population as $\boldsymbol{x}_{i,G+1}$. DE is quite robust, fast, and effective, with global optimization ability. It does not require the objective function to be differentiable, and it works well even with noisy and time-dependent objective functions.

Description of the used DERand1Bin strategy (both for Chaos DE and Canonical DE) is presented in (3). Please refer to [23] and [24] for the detailed complete description of all other strategies.

$$u_{j,i,G+1} = x_{j,r1,G} + F \cdot (x_{j,r2,G} - x_{j,r3,G}) \tag{3}$$

5 Lozi Map

This section contains the description discrete chaotic map used as the chaotic pseudo random generator for DE. In this research, direct output iterations of the chaotic map were used for the generation of real numbers in the process of crossover based on the user defined *CR* value and for the generation of the integer values used for selection of individuals. The initial concept of embedding chaotic dynamics into evolutionary algorithms is given in [25].

The Lozi map is a simple discrete two-dimensional chaotic map. The map equations are given in Eq. (4) and (5). The parameters used in this work are: $a = 1.7$ and $b = 0.5$ as suggested in [26]. For these values, the system exhibits typical chaotic behavior and with this parameter setting it is used in the most research papers and other literature sources. The Lozi map is given in Fig. 2.

$$X_{n+1} = 1 - a\,|X_n| + bY_n \tag{4}$$

$$Y_{n+1} = X_n \tag{5}$$

6 Problem Design

The cost function (CF) that was minimized is given in (6). It is divided into three time intervals, and has two penalizations. The first part ensures minimizing the area between required and actual temperature of the reaction mixture, and the second part ensures the rapid reaction of the whole batch of chemical FK, thus the very low value of concentration a_{FK} of partly unreacted portions of

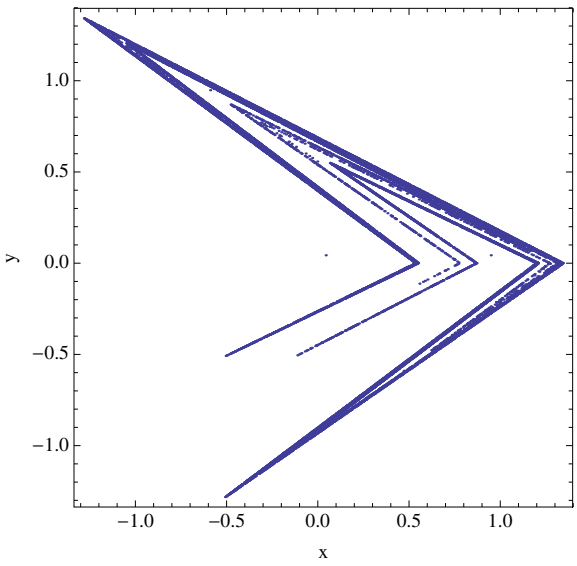


Fig. 2. Lozi map

chemical FK in reaction mixture. The first penalization helps to find solutions in which the temperature of reaction mixture cools down fast to its initial state, and the process duration is shortened. The second corresponds to the critical temperature.

$$f_{\cos t} = \sum_{t=0}^{t_1} |w - T[t]| + \sum_{t=0}^{t_1} a_{FK}[t] + pen.1 + pen.2 \tag{6}$$

$$pen.1 = \begin{cases} 0 & Max\,(T[\tau]) \leq 323,15 \\ 50000 & else \end{cases}$$

$$\text{for } \tau \in \langle t_2, t_3 \rangle$$

$$pen.2 = \begin{cases} 0 & Max\,(T[\tau]) \leq 373,15 \\ 50000 & else \end{cases}$$

$$\text{for } \tau \in \langle 0, t_3 \rangle$$

Where the time intervals were set for example as: t_1= 15000 s; t_2=20000 s; t_3=25000 s.

The minimizing term, presented in (7), limits the maximum mass of one batch. Moreover, many parameters were interrelated due to the optimization of the reactor geometry.

$$m[t] \leq m_{\max} \tag{7}$$

7 Optimization Results

The parameter settings for both Chaos DE and Canonical DERand1Bin strategy were obtained analytically based on numerous experiments and simulations (see Table 1). Experiments were performed in an environment of Wolfram Mathematica and were repeated 100 times.

Table 1. Parameter set up for Chaos DE and Canonical DE

Parameter	Value
PopSize	50
F	0.8
CR	0.8
Generations	200
Max. CF Evaluations (CFE)	10000

The optimization proceeded with the parameters shown in Table 2, where the internal radius of reactor is expressed in parameter r and is related to cooling area S. Parameter d represents the distance between the outer and inner jackets and parameter h means the height of the reactor.

Table 2. Optimized reactor parameters, difference between original and the optimized reactor

Parameter	Range	Original value	Optimized value
$\dot{m}_{FK}$ [kg.s^{-1}]	0 – 10.0	0 - 3	0,0693
T_{VP} [K]	273.15 – 323.15	293.15	276.825
$\dot{m}_V$ [kg]	0 – 10.0	1	4,9505
r [m]	0.5 – 2.5	0.78	1.0004
h [m]	0.5 – 2.5	1.11	1.0026
d [m]	0.01 – 0.2	0.03	0.0389

The original design of the reactor was based on standard chemical-technological methods and gives a proposal of reactor physical dimensions and parameters of chemical substances. These values are called within this paper **original parameters (values)**. The best results of the optimization are shown in Fig. 3 and Table 2. From these results it is obvious that the temperature of reaction mixture did not exceed the critical value. The maximum temperature was 373.11 K (99.95°C). The required temperature w used in cost function was 370.0 K (96.85°C). Another fact not to be neglected is the shortened duration of the process (approx 20670 s compared to the original approx. 25000 s).

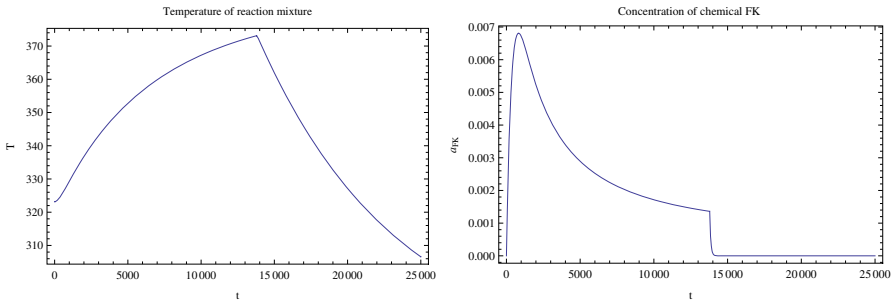


Fig. 3. Results of the optimization. Simulation of the **best solution**, course of the temperature of reaction mixture (left) and concentration of chemical FK (right).

8 Performance Comparison for Chaos DE and Canonical DE

The performance comparison of Chaos DE with Lozi map against Canonical DERand1Bin strategy is given in Tables 3 and 4 and Figures 4 – 6. Table 3 contains the important statistical values for the results of all 100 runs and Table 4 compares the progress of Chaos De and Canonical DE. The Table 4 contains the average CF values for the generation No. 50, 100, 150 and 200 from all 100 runs. The Fig. 4 shows the comparison of the time evolution of CF values for the best individual solutions, which are the solutions with the lowest CF values from all 100 runs of either chaos DE or canonical DE, whereas Fig. 5 shows the time evolution of CF values for the best progressive individual solutions. These individual solutions represent the ones with the lowest sum of the CF values with the step of 20 generations, i.e. with the best progress towards the global optimum. The Fig. 6 represents the comparison of the time evolution of average CF values from all 100 runs. Finally the Fig. 7 confirms the robustness of Chaos DE in finding the best solutions for all 100 runs.
Based on the presented graphical results and statistical analysis it is possible to claim that embedding the chaotic dynamics in the form of chaotic pseudo random number generator into differential evolution algorithm significantly improves the performance of the DE.

Table 3. Statistical results of all 100 runs of Chaos DE and Canonical DE

DE Version	Avg. CF	Median CF	Std. Dev.	Max CF	Min CF
Canonical DE	21259.0	21234.1	94.1	21588.2	21089.2
Chaos DE Lozi	21107.6	21107.4	25.4	21160.7	21059.0

Table 4. Average CF values for the generation No. 50, 100, 150 and 200 from all 100 runs of Chaos DE and Canonical DE

Generation No.:	50	100	150	200
Avg. CF for Canonical DE	25320.1	22719.9	21619.8	21259.0
Avg. CF for Chaos DE Lozi	24707.1	21992.1	21265.2	21107.6

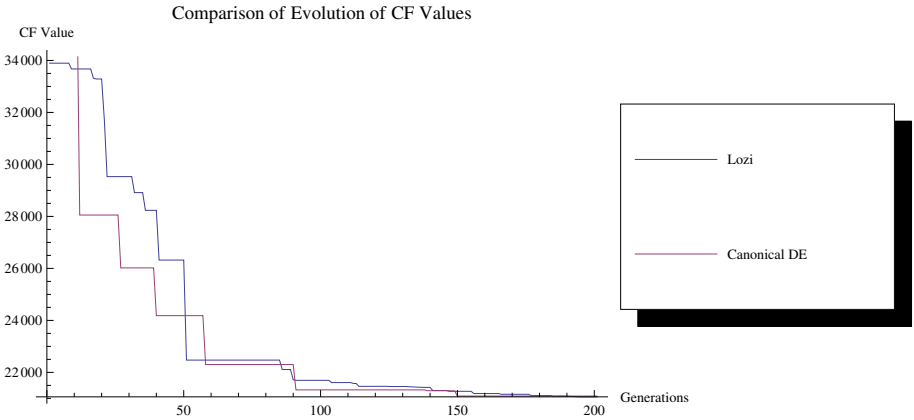


Fig. 4. Evolution of CF values for the best individual solutions

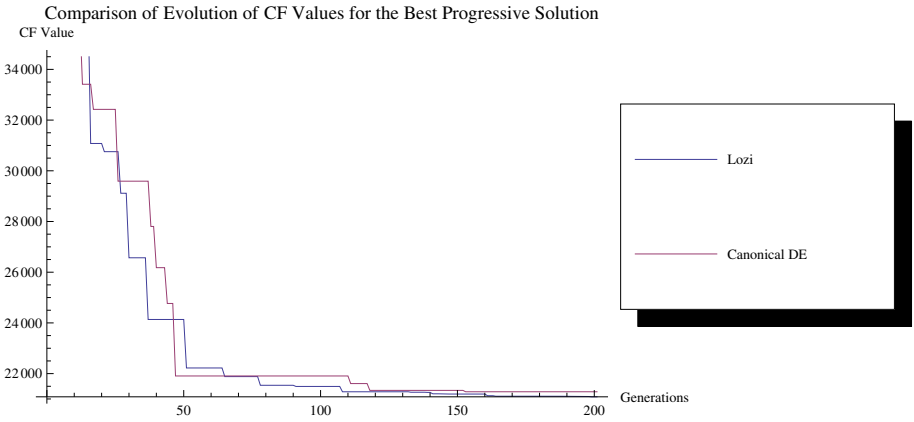


Fig. 5. Evolution of CF values for the best progressive individual solutions

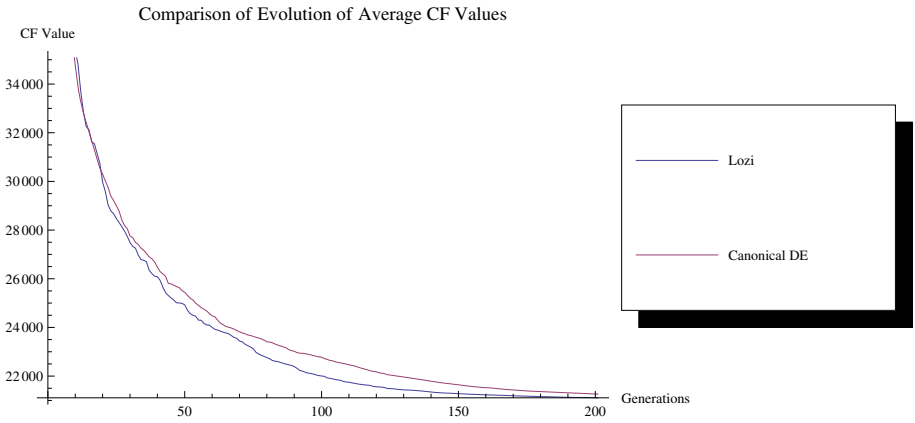


Fig. 6. Evolution of average CF values for all 100 runs

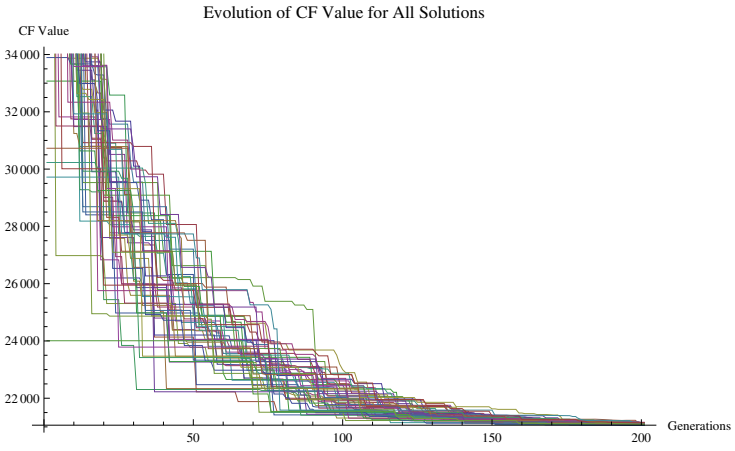


Fig. 7. Evolution of CF value for all 100 runs of Chaos-DE Lozi

9 Conclusions

Based on obtained results, it may be claimed, that the presented Chaos DE driven by means of the chaotic Lozi map has given satisfactory results. The behavior of an uncontrolled original reactor gives quite unsatisfactory results. The reactor parameters found by optimization brought performance superior compared to the reactor set up by an original parameters.

Moreover the results of the performance comparison against the canonical DE show that embedding of the chaotic dynamics in the form of chaotic pseudo random number generator into the differential evolution algorithm significantly improves the performance of the DE.

Future plans include testing of different chaotic systems, comparison with different heuristics and obtaining a large number of results to perform statistical tests.

References

1. He, Q., Wang, L.: An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Engineering Applications of Artificial Intelligence 20, 89–99 (2007)
2. Silva, V.V.R., Khatib, W., Fleming, P.J.: Performance optimization of gas turbine engine. Engineering Applications of Artificial Intelligence 18, 575–583 (2005)
3. Tan, W.W., Lu, F., Loh, A.P., Tan, K.C.: Modeling and control of a pilot pH plant using genetic algorithm. Engineering Applications of Artificial Intelligence 18, 485–494 (2005)
4. Lepore, R., Wouwer, A.V., Remy, M., Findeisen, R., Nagy, Z., Allger, F.: Optimization strategies for a MMA polymerization reactor. Computers and Chemical Engineering 31, 281–291 (2007)
5. Rout, B.K., Mittal, R.K.: Optimal manipulator parameter tolerance selection using evolutionary optimization technique. Engineering Applications of Artificial Intelligence 21, 509–524 (2008)
6. Davendra, D., Zelinka, I., Senkerik, R.: Chaos driven evolutionary algorithms for the task of PID control. Computers & Mathematics with Applications 60(4), 1088–1104 (2010) ISSN 0898-1221
7. Senkerik, R., Davendra, D., Zelinka, I., Oplatkova, Z., Pluhacek, M.: Optimization of the Batch Reactor by Means of Chaos Driven Differential Evolution. In: Snasel, V., Abraham, A., Corchado, E.S. (eds.) SOCO Models in Industrial & Environmental Appl. AISC, vol. 188, pp. 93–102. Springer, Heidelberg (2013) ISBN: 978-3-642-32922-7
8. Pluhacek, M., Budikova, V., Senkerik, R., Oplatkova, Z., Zelinka, I.: On the Performance of Enhanced PSO Algorithm with Lozi Chaotic Map – an Initial Study. In: Proceedings of the 18th International Conference on Soft Computing, MENDEL 2012, pp. 40–45 (2012) ISBN 978-80-214-4540-6
9. Silva, C.M., Biscaia, E.C.: Genetic algorithm development for multi-objective optimization of batch free-radical polymerization reactors. Computers and Chemical Engineering 27, 1329–1344 (2003)
10. Arpornwichanop, A., Kittisupakorn, P., Mujtaba, M.I.: On-line dynamic optimization and control strategy for improving the performance of batch reactors. Chemical Engineering and Processing 44(1), 101–114 (2005)
11. Srinisavan, B., Palanki, S., Bonvin, D.: Dynamic optimization of batch processes I. Characterization of the nominal solution. Computers and Chemical Engineering 27, 1–26 (2002)
12. Srinisavan, B., Palanki, S., Bonvin, D.: Dynamic optimization of batch processes II. Role of Measurement in handling uncertainly. Computers and Chemical Engineering 27, 27–44 (2002)
13. Wang, Y., Zhou, D., Gao, F.: Iterative learning model predictive control for multiphase batch processes. Journal of Process Control 18, 543–557 (2008)
14. Cho, W., Edgar, T.F., Lee, J.: Iterative learning dual-mode control of exothermic batch reactors. Control Engineering Practice 16, 1244–1249 (2008)

15. Beyer, M.A., Grote, W., Reinig, G.: Adaptive exact linearization control of batch polymerization reactors using a Sigma-Point Kalman Filter. Journal of Process Control 18, 663–675 (2008)
16. Sarma, P.: Multivariable gain-scheduled fuzzy logic control of an exothermic reactor. Engineering Applications of Artificial Intelligence 14, 457–471 (2001)
17. Sberg, J., Mukul, A.: Trajectory tracking in batch processes using neural controllers. Engineering Applications of Artificial Intelligence 15, 41–51 (2002)
18. Mukherjee, A., Zhang, J.: A reliable multi-objective control strategy for batch processes based on bootstrap aggregated neural network models. Journal of Process Control 18, 720–734 (2008)
19. Mujtaba, M., Aziz, N., Hussain, M.A.: Neural network based modelling and control in batch reactor. Chemical Engineering Research and Design 84(8), 635–644 (2006)
20. Causa, J., Karer, G., Nunez, A., Saez, D., Skrjanc, I., Zupancic, B.: Hybrid fuzzy predictive control based on genetic algorithms for the temperature control of a batch reactor. Computers and Chemical Engineering (2008), doi:10.1016/j.compchemeng, 05.014
21. Altinten, A., Ketevanlioglu, F., Erdogan, S., Hapoglu, H., Alpbaz, M.: Self-tuning PID control of jacketed batch polystyrene reactor using genetic algorithm. Chemical Engineering Journal 138, 490–497 (2008)
22. Faber, R., Jockenhel, T., Tsatsaronis, G.: Dynamic optimization with simulated annealing. Computers and Chemical Engineering 29, 273–290 (2005)
23. Price, K.: An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 79–108. McGraw-Hill, London (1999) ISBN 007-709506-5
24. Price, K., Storn, R.: Differential evolution homepage (2001), `http://www.icsi.berkeley.edu/~storn/code.html`
25. Caponetto, R., Fortuna, L., Fazzino, S., Xibilia, M.: Chaotic sequences to improve the performance of evolutionary algorithms. IEEE Trans. Evol. Comput. 7(3), 289–304 (2003)
26. Sprott, J.C.: Chaos and Time-Series Analysis. Oxford University Press (2003)

Application of Geometric Differential Evolution Algorithm to Design Minimal Phase Digital Filters with Atypical Characteristics for Their Hardware or Software Implementation

Adam Slowik

Department of Electronics and Computer Science,
Koszalin University of Technology, Sniadeckich 2 Street,
75-453 Koszalin, Poland
aslowik@ie.tu.koszalin.pl

Abstract. In this paper, the application of a geometric differential evolution algorithm to design minimal phase digital filters with atypical characteristics is presented. Owing to the method proposed, we can design digital filters for any numerical systems in dedicated hardware implementation. Moreover, with the use of a geometric differential evolution algorithm, we can create digital filters for hardware and/or software implementation using the same design algorithm. In the paper, a design of two digital filters in the Q.15 numerical format (for hardware realization) and in the real number numerical format (for software realization) is presented. The results obtained using the proposed method are better than the results obtained with the use of the other methods.

1 Introduction

An efficient design of digital filters with atypical characteristics is more and more important in today's world [1]. These days, digital filters with atypical characteristics are needed in any equalizers, any DSP hearing aid systems, any phase or amplitude correctors, and in any specialized audiophile systems. If we wish to design a digital filter with standard amplitude characteristics, we may use one of the following approximations, e.g.: Butterworth, Chebyshev, Cauer. However, when the filter designed must possesses atypical amplitude characteristics, we can use the Yule-Walker (YW) method [2, 3] or any global optimization method which is dedicated to the digital filter design.

The YW method designs recursive infinite impulse response (IIR) digital filters using the least-squares fit to a specified frequency response [2, 3]. In literature, we can also find some optimization algorithms which are suitable for this problem. As an example, we may mention the following: genetic algorithms (GA) [4-7], ant colony optimization algorithms [8, 9], particle swarm optimization algorithms [10-12], and any other techniques which are based on nature. Sometimes, when we are not quite certain whether the set of admissible solutions in the search space is non-empty, as an examination of the problem solvability,

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 67–78, 2013.

we can use constraint programming methods [22, 23] before the application of the adequate optimization algorithm. The problem of the digital filter design with atypical amplitude characteristics is more complicated if we wish to design a digital filter which will be suitable for a given numerical system in a dedicated hardware implementation. In this case, the application of the YW method does not constitute a good solution because the results (filter coefficients) obtained using the YW method must be scaled and quantized before any hardware implementation of the digital filter designed. The quantizing process can introduce some additional errors for the hardware implementation of a given digital filter [13]. Therefore, as a result we can obtain a hardware digital filter with slightly different properties than its software version obtained using the YW method. Especially, it is very important if the filter designed is a IIR digital filter as IIR digital filters are very sensitive to any changes of the filter coefficient values.

In order to find a solution to this problem, some researchers use genetic algorithms to design digital filters [1, 14-17] or a hybrid method based on a genetic algorithm with initialization by a scaled and quantized solution obtained using the YW method [13]. In both of these cases, we have a discrete optimization problem and it is only discrete global optimization methods that can be used. Yet, what about the case when we wish to have one optimization method for the design of digital filters for software implementation (a continuous optimization problem) and for the design of digital filters for hardware implementation (a discrete optimization problem). Generally, if we are based on the genetic algorithm we must create two stand-alone optimization methods. One for software implementation of given filters, and one for hardware implementation of given filters. These two methods are necessary because different crossover operators and mutation operators are required for continuous and discrete problems which are solved using GA.

In this paper, we wish to overcome these disadvantages and we intend to propose one efficient method which can be simultaneously used for designing digital filters for software implementation and for designing digital filters for hardware implementation. Our main goal is to propose an application of a geometric differential evolution algorithm [18] to design minimal phase IIR digital filters with atypical characteristics for various numerical systems in a dedicated hardware and/or software implementation. The method presented in this paper is known as GDEFD (Geometric Differential Evolution for Filter Design).

The presentation of the paper is as follows: in Section Two we present a brief review on the Geometric Differential Evolution algorithm, in Section Three we present a brief review on IIR digital filters, in Section Four we present the proposed method (GDEFD), in Section Five we present an objective function for our GDEFD algorithm, in Section Six we present some experimental results, and finally in Section Seven some conclusions are presented.

2 Geometric Differential Evolution Algorithm

The Geometric Differential Evolution (GDE) algorithm [18] is a recently introduced formal generalization of the traditional Differential Evolution (DE) [19, 20]

algorithm that applies naturally to both continuous and combinatorial spaces [18]. The formal version of the GDE algorithm [18] is presented in Algorithm 1 (*OF(.)* is the objective function; algorithm minimize *OF(.)*)

Algorithm 1. Geometric Differential Evolution Algorithm

initialize population of $PopSize$ individuals at random (each individual consists of N values; N is equal to the space dimension being optimized)
while stop criteria not met **do**
    for all individuals $X(i)$ in the population **do**
        chose at random three distinct individuals $(X1, X2, X3)$ from the current population
        set $W = 1/(1 + F)$ where F is a scale parameter
        create intermediate individual E as the convex combination $CX(X1, X3)$ with weights $(1 - W, W)$; generalizing $E = (1 - W) \cdot X1 + W \cdot X3$
        create mutant individual U as the extension ray $ER(X2, E)$ with weights $(W, 1 - W)$; generalizing $U = (E - (1 - W) \cdot X2)/W$
        create candidate individual V as the convex combination $CX(U, X(i))$ with weights $(Cr, 1 - Cr)$ where Cr is recombination parameter; generalizing $V = Cr \cdot U + (1 - Cr) \cdot X(i)$
        if $OF(V) \geq OF(X(i))$ **then**
            set the $i - th$ individual in the next population $Y(i) = V$
        else
            set $Y(i) = X(i)$
        end if
    end for
    for all individual $X(i)$ in the population **do**
        set $X(i) = Y(i)$
    end for
end while

3 Infinite Impulse Response Digital Filters

Generally, the transfer function of the designed IIR digital filter in z domain is described as follows [21]:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + ... + b_{n-1} \cdot z^{-(n-1)} + b_n \cdot z^{-n}}{1 - \left(a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + ... + a_{n-1} \cdot z^{-(n-1)} + a_n \cdot z^{-n}\right)} \quad (1)$$

where: a_i, b_i are filter coefficients, z^{-1} is a delay element, *X(z)* is a vector of input samples in z domain, *Y(z)* is a vector of output samples in z domain, n is a filter order.

It is evident based on Equation 1 that the value of a_0 coefficient is equal to 1. It is true in most cases of the IIR digital filter design, but in this paper we assume that value a_0 is a parameter because if the filter coefficients obtained using the YW method are scaled and quantized, the value of a_0 is frequently not

equal to 1. Therefore, in our approach, the transfer function of IIR filters in z domain can be described using the following equation:

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + ... + b_{n-1} \cdot z^{-(n-1)} + b_n \cdot z^{-n}}{a_0 - \left(a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + ... + a_{n-1} \cdot z^{-(n-1)} + a_n \cdot z^{-n}\right)} \tag{2}$$

The main goal of the design algorithm of digital filters is to find such a set of filter coefficients a_i, b_i ($i \in [0; n]$, where n is a filter order) in order to assure the stability of the filter designed, the minimal phase of the filter, and to fulfill all design-related assumptions. However, if we wish to obtain a hardware realization of the digital filter, which will be resistive to rounding errors, the filter coefficients must accept exactly the values determined dependent on the number of bits used in the representation of each filter coefficient. In the case when the number of bits which are used to represent the filter coefficient is equal to nb, the M=nb-1 bits are allowable in the realization of its value (one bit is taken as a sign bit). Therefore, the digital filter coefficients can accept the values from the following domain D (in fixed-point format Q.M):

$$D = \left[\frac{(-1) \cdot 2^M}{2^M}; \frac{2^M - 1}{2^M}\right] \tag{3}$$

In the 2's complement fractional representation, an nb bit binary word can represent 2^{nb} equally spaced numbers from $\frac{(-1)\cdot 2^M}{2^M} = -1$ to $\frac{2^M-1}{2^M} = 1 - 2^{-M}$ (see equation 3).

The binary word BW which consists of nb bits (bw_i): BW=bw_M, bw_{M-1}, bw_{M-2}, ..., bw_2, bw_1, bw_0 we interpret as a fractional number x:

$$x = -(b_M) + \sum_{i=0}^{M-1} \left(2^{i-M} \cdot bw_i\right) \tag{4}$$

Of course if we use a fractional number in Q.M format, the value of coefficient a_0 will be not equal to 1, but a_0 will be equal to $1 - 2^{-M}$.

If we design a digital filter for software implementation, the coefficient values constitute the real number from range $[-1; +1]$.

In order to assure the stability of digital filters, the poles of the function (2) must be placed in an unitary circle in the z plane. Of course, in order to assure that the designed filter is a minimal phase digital filter, the zeros of the function (2) must be placed also in the unitary circle in the z plane.

4 Geometric Differential Evolution for Filter Design Method

The proposed GDEFD (Geometric Differential Evolution for Filter Design Method) is based on the ninth step.

For hardware implementation of the digital filter designed
In the first step, the set D consisting of 2^{nb} values is created (see Equation 3; of course, this equation is true in the case when we design a digital filter in Q.15 numerical format; if the numerical format for given DSP system is other than Q.15 format, we must create the set D using an adequate formula for the numerical format in our DSP system). For each value from the set D, the index value is assigned. The first value from set D possesses index number 1, the last value from the same set is represented by index 2^{nb} (in our case, nb is equal to 16 bits, therefore we have 65536 elements in the set D). Next, the population *Pop* is randomly created. The population *Pop* consists of *PopSize* individuals. Each individual in the population consists of $2 \cdot (n + 1)$ genes (n represents the filter order; in our case, we have assumed that the filter order n is equal to 10). Each gene in the individual accepts one integer value from the range $[1; 2^{nb}]$ (in our case, the range is [1; 65536]). The value recorded in each gene indicates the adequate filter coefficient from the previously created set D.

For software implementation of the digital filter designed
In the first step, the population *Pop* is randomly created. The population *Pop* consists of *PopSize* individuals. Each individual in the population consists of $2 \cdot (n + 1)$ genes (n that represent the filter order; in our case, we have assumed that the filter order n is equal to 10). Each gene in the individual accepts one real value from the range $[-1; +1]$. The value which is recorded in a given gene represents the suitable coefficient of the digital filter designed.

For hardware implementation of the digital filter designed
In the second step, a given digital filter is designed using the YW method. The filter coefficients obtained using the YW method (for the given digital filter) are scaled into the range $[-1; +1]$. The scaling process is described by the following example. Let us assume that after using the YW method (for the 4-th order IIR digital filter), we obtained the filter coefficients a_i and b_i as follows:

$$a_0 = 1; a_1 = 3.2145; a_2 = -2.7810; a_3 = -1.2098; a_4 = -0.0012;$$

$$b_0 = 0.2345; b_1 = -0.4345; b_2 = 1.0120; b_3 = 0.1121; b_4 = 1.0322;$$

When we possess the above mentioned filter coefficients, we search the highest absolute value from these coefficients (a_i and b_i). We may see that in our example, the highest absolute value is equal to $|a_1| = 3.2145$. Next, we search the lowest value of the power of number 2 which is higher than the highest absolute value from the filter coefficients; in our example, the lowest value of the power of number 2 is equal to 4. Finally, all the filter coefficient values are divided by the lowest value of the power of number 2 which was found; in our example, all the filter coefficient values are divided by 4. Due to this divide, we obtained new filter coefficient values which are from the range $[-1; +1]$, and are as follows:

$$a'_0 = 0.2500; a'_1 = 0.8036; a'_2 = -0.6953; a'_3 = -0.3025; a'_4 = -0.0003;$$

$$b'_0 = 0.0586; b'_1 = -0.1086; b'_2 = 0.2530; b'_3 = 0.0280; b'_4 = 0.2581;$$

Also, we wish to observe that the scaling process does not introduce any additional errors; the digital filter with the values of coefficients before and after scaling process possesses the same properties from the objective function point of view (see Section 5 where the objective function $OF(.)$ is described in detail).

Next, these scaled filter coefficients are quantized into a fixed-point format selected by the user (in our case, into Q.15 format). At the end of this step, the scaled and quantized filter coefficients are written down into randomly selected individuals from the population *Pop*.

For software implementation of the digital filter designed
In the second step, the given digital filter is designed using the YW method. The filter coefficients obtained using the YW method (for the given digital filter) are scaled into the range $[-1; +1]$. The scaling process was described in the previous paragraph of this paper.

At the end of this step, the scaled values of digital filter coefficients are written down into randomly selected individuals from the population *Pop*.

For hard- and software implementation of the digital filter designed
In the third step, an evaluation of all the individuals using the objective function *OF(.)* is performed (the objective function *OF(.)* is described in Section Five in this paper). The GDEFD algorithm presented tends to minimize the objective function *OF(.)*.

In the fourth step, the best individual (one possessing the lowest value of the objective function *OF(.)*) is selected from the current population *Pop*. In the first algorithm iteration, the best individual selected is remembered in the variable *TheBest*. During further algorithm iterations, the best individual selected is remembered as the *TheBest*, if and only if the value of its objective function *OF(.)* is lower than the value of the objective function for the individual actually stored in the variable *TheBest*.

In the fifth step, for each $i-th$ individual $X(i)$ in the population, we chose three individuals $TheBest$, $X2$, and $X3$ (these individials must be different; the individual $X2$ and $X3$ are chosen randomly). Next, for each individual in the population, we compute the value F according to the following formula:

$$F = \frac{TheBest.eval_g + \epsilon}{TheBest.eval_{g-1} + \epsilon} \tag{5}$$

where: $TheBest.eval_g$ is the value of the objective function for the individual stored in the variable $TheBest$ during the current $g-th$ algorithm generation, $TheBest.eval_{g-1}$ is the value of the objective function for the individual stored in the variable $TheBest$ during the previous $(g-1)-th$ algorithm generation, *random* is a random value from the range $[0; 1)$, ϵ is a very small positive value;

due to the ϵ value, the problem of the division by zero was eliminated in Equation (5); in our algorithm, we assumed that the ϵ value is equal to 0.00001.

Equation (5) constitutes our approach for a better search of the solution space by the GDE (Geometric Differential Evolution) algorithm. If we have a stagnation (the best solutions obtained by the algorithm in the successive generations possess the same value of the objective function), then the value of the F parameter is equal to 1 (due to this the value, W is equal to 0.5; cf. Equation 6). In other cases, the value of the F parameter is equal to a value being lower than 1, which is related to the changes of the best solution value in the population. Also, in this step, for each individual in the population, the W value is computed according to the following formula:

$$W = \frac{1}{1+F} \tag{6}$$

In the sixth step, the individuals $E1$, U and V are computed for each individual in the population according to the following formula:

$$E1 = (1-W) \cdot TheBest + (W \cdot X3) \tag{7}$$

$$U = \frac{(E1 - ((1-W) \cdot X2))}{W} \tag{8}$$

$$V = round\left((Cr \cdot U) + (1-Cr) \cdot X\left(i\right)\right) \tag{9}$$

where: *round* is the function for the rounding of the real value to the integer value, Cr is a crossover rate (in our algorithm, we assume that Cr is equal to 0.5).

For hardware implementation of the digital filter designed
In the seventh step, the values stored in the individual V are rounded to the nearest integer value (we use *round* command which is availabe in all programming languages). Next, we check the values of the genes in the individual V. If the value stored in a given gene is higher than 2^{nb} (in our case higher than 65536) or lower than 1, the new integer value of the gene is randomly chosen from the range [1; 2^{nb}] (in our case, from the range [1; 65536]).

For software implementation of the digital filter designed
In the seventh step, we check the values of the genes in the individual V. If the value stored in the given gene is higher than $+1$ or lower than -1, then the new value of the gene is randomly chosen from the range $[-1; +1]$.

For hard- and software implementation of the digital filter designed
In the eight step, the individual V for each individual from the population is evaluated using an objective function. If the value of the objective function for the V individual is lower than the value of the objective function for the individual $X(i)$, the individual $X(i)$ is replaced by the individual V in the population. In other cases, in the population, the individual $X(i)$ is not replaced by the individual V.

In the ninth step, the termination criteria are verified. In our case, the termination criterion is the number of the generation G_{max}. If the value of the algorithm iteration is higher than the value of G_{max}, the algorithm is stopped. As a result, we obtain filter coefficients that are stored in the $TheBest$ individual. In other cases, the algorithm is to jump to step three.

5 Objective Function OF(.)

The objective function *OF(.)* is computed as follows (in equations (10-16) the index i represents i-th individuals in population):

$$FC_i = AmplitudeError_i + w \cdot (StabError_i + MinPhaseError_i) \tag{10}$$

$$AmplitudeError_i = \sum_{k=1}^{R} AmpErr_{i,k} \tag{11}$$

$$AmpErr_{i,k} = \begin{cases} H(f_k)_i - Upper_{i,k}, & when H(f_k)_i > Upper_{i,k} \\ Lower_{i,k} - H(f_k)_i, & when H(f_k)_i < Lower_{i,k} \\ 0, & otherwise \end{cases} \tag{12}$$

$$StabError_i = \sum_{j=1}^{J} StabErr_{i,j} \tag{13}$$

$$StabErr_{i,j} = \begin{cases} |p_{i,j}| - 1, & when\ |p_{i,j}| \geq 1 \\ 0, & otherwise \end{cases} \tag{14}$$

$$MinPhaseError_i = \sum_{q=1}^{Q} PhaseErr_{i,q} \tag{15}$$

$$PhaseErr_{i,q} = \begin{cases} |z_{i,q}| - 1, & when\ |z_{i,q}| \geq 1 \\ 0, & otherwise \end{cases} \tag{16}$$

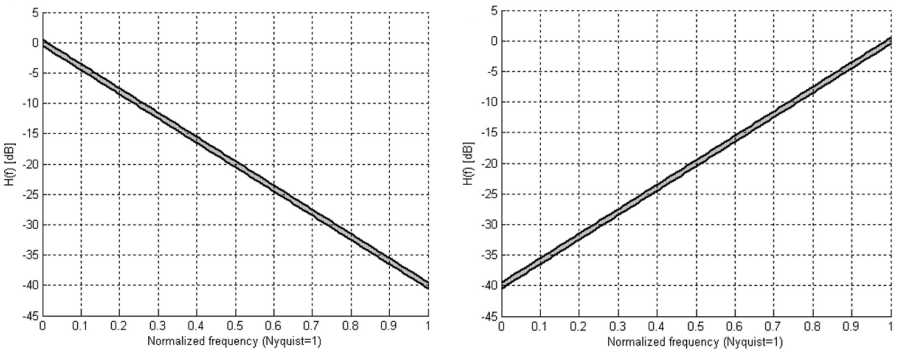
where: w is a value of penalty factor (during experiments $w = 10^5$ is assumed; the value of w has been taken experimentally, but in general, the value of w must be higher, than the highest value which can be obtained in *AmplitudeError*), $AmpErr_{i,k}$ is a partial value of amplitude characteristics error for k-th value of normalized frequency, $H(f_k)_i$ is a value of amplitude characteristics for k-th value of normalized frequency f, $Lower_{i,k}$ is a value of lower constraint for amplitude characteristics value for k-th value of normalized frequency, $Upper_{i,k}$ is a value of upper constraint for amplitude characteristics value for k-th value of normalized frequency, $StabErr_{i,j}$ is a partial filter stability error for j-th pole of transmittance function, J is a number of poles of transmittance function, $|p_{i,j}|$ is a value of module for j-th pole of transmittance function, $PhaseErr_{i,q}$ is a partial filter minimal phase error for q-th zero of transmittance function, Q is

a number of zeros of transmittance function, $|z_{i,q}|$ is a value of module for q-th zero of transmittance function.

In order to obtain the value of objective function *OF(.)* for i-th individual in the population, first the amplitude characteristics $H(f)_i$ which coefficients are stored in the i-th individual is computed. The amplitude characteristics is computed using R values of normalized frequency $f \in [0; 1]$ (where 1 represents the Nyquist frequency; in proposed method normalized frequency is divided into R points). Also, the poles and zeros of transmittance function (see equation 2) are computed for each individual in the population *Pop*. When we have amplitude characteristics and the values of poles and zeros of the transmittance function for i-th individuals, we can compute the objective function *OF(.)*.

6 Experimental Verification of Proposed Method

In order to test the proposed GDEFD method, two minimal phase IIR digital filters were designed. The first digital filter possesses linearly falling amplitude characteristics, and the second one possesses linearly growing amplitude characteristics. During the experiments, we changed the values of the deviations of the obtained amplitude characteristics from the ideal (assumed) amplitude characteristics. In Figure 1, we show the shape and detailed information concerning the amplitude characteristics of the exemplary digital filters designed.



(a) (b)

Fig. 1. Shape of amplitude characteristics of designed minimal phase IIR digital filters: linearly falling (a), linearly growing (b)

The parameters for the proposed GDEFD methods are as follows: the filter order $n = 10$, the number of individuals in the population $PopSize = 100$, the crossover rate $Cr = 0.5$, the maximal number of algorithm generations $G_{max} = 1000$; during the experiments, the normalized frequency is divided into 128 points ($R = 128$). In Table 1, the results obtained using a GDEFD

method and other methods are presented. The symbols in Table 1 are as follows: *Dev [dB]* is the accepted maximal deviation of the amplitude characteristics, *GDEFD for Hardware* represents the results obtained using the proposed method for hardware implementation of the digital filter designed, *GDEFD for Software* represents the results obtained using the proposed method for software implementation of the digital filter designed, *Best* is the best value and $Worst$ is the worst value of the objective function *OF(.)* obtained after a 10-fold repetition of the proposed algorithm, *Avg* is the average value and *StdDev* is the standard deviation value of the objective function *OF(.)* obtained after a 10-fold repetition of the proposed algorithm, $YWSQ$ is the scaled and quantized result obtained using the Yule-Walker method, YW is the result obtained using the Yule-Walker method. The value of the objective function *OF(.)* equal to 0 represents the digial filter for which all the design assumptions and constraints were fulfilled.

Table 1. The results obtained when designing the minimal phase IIR digital filter with linearly falling and linearly growing amplitude characteristics using different design methods

Minimal phase IIR digital filter with linearly falling amplitude characteristics										
Dev	GDEFD for Hardware				GDEFD for Software				YWSQ	YW
[dB]	Best	Avg	StdDev	Worst	Best	Avg	StdDev	Worst		
0.0	4.4943	6.4683	2.0159	11.3074	3.1758	3.6124	0.2753	4.0976	21.1348	4.3616
0.1	0.7040	1.9873	0.9545	3.4218	0.1854	0.4903	0.2031	0.7304	17.4830	1.1747
0.2	0.3497	2.7424	2.5487	7.5902	0	0.2076	0.1609	0.4221	16.3572	0.7172
0.3	0.2522	1.2597	1.3444	4.6930	0	0.0714	0.0643	0.1398	15.4048	0.4074
0.4	0	0.7300	1.2448	3.9936	0	0.0003	0.0008	0.0023	14.5563	0.1860
0.5	0	2.5435	4.4432	13.7978	0	0	0	0	13.7978	0.0811
Minimal phase IIR digital filter with linearly growing amplitude characteristics										
Dev	GDEFD for Hardware				GDEFD for Software				YWSQ	YW
[dB]	Best	Avg	StdDev	Worst	Best	Avg	StdDev	Worst		
0.0	3.8006	12.6551	4.3078	17.4019	10.9771	12.4834	0.6387	12.9582	45.8526	13.3234
0.1	1.3981	2.6831	1.0443	4.7664	0.4800	1.0874	0.3205	1.2962	33.9491	1.4024
0.2	0.0749	0.3917	0.3294	1.1502	0	0.3788	0.2694	0.6949	32.0584	0.8079
0.3	0	0.8505	1.5333	5.0024	0	0.1408	0.1841	0.4689	30.6279	0.5048
0.4	0	0.6982	2.0680	6.5791	0	0.1024	0.1222	0.2493	29.5510	0.3048
0.5	0	0.6182	0.9152	2.3393	0	0.0109	0.0326	0.0977	28.6575	0.1840

It can be seen in Table 1 that results obtained using the proposed GDEFD method (for Hardware and Software implementation of the digital filter designed) are better than the results obtained using the YWSQ and YW methods.

In order to admit an importance of result obtained using both versions of GDEFD method on the background of the results obtained using YW method (for GDEFD method for Software implementation of digital filters) and YWSQ method (for GDEFD method for Hardware implementation of digital filters),

t-Student statistical test (with 18 degrees of freedom) has been performed. The obtained results are not presented in this paper with respect to space limitation. But in all cases, the results obtained using both versions of proposed GDEFD method are statistically important (with 99% degree of trust).

Also, it must be admitted that the computational time for the method proposed is much higher than the computational time required for the YW method. The results including a comparison of computational times required for all the methods are not presented in this paper for space limitation considerations.

7 Conclusions

In the paper, a method for designing minimal phase IIR digital filters with atypical amplitude characteristics for hardware and software realization was presented. The main advantage of the method proposed is that the filter designed can be directly implemented into the hardware without any additional errors. Using the method proposed, minimal phase IIR digital filters can be designed for any numerical format which is for example available on the target DSP system. The digital filters designed using the GDEFD method possess better properties than those digital filters that are designed using the YWSQ and YW methods. Also, it is worth to notice that when using the proposed algorithm (GDEFD), we can design minimal phase IIR digital filters for hardware and/or software implementation simultaneously (using the same and only one design algorithm).

References

1. Slowik, A.: Application of Evolutionary Algorithm to Design of Minimal Phase Digital Filters with Non-Standard Amplitude Characteristics and Finite Bits Word Length. Bulletin of The Polish Academy of Science - Technical Science 59(2), 125–135 (2011), doi:10.2478/v10175-011-0016-z
2. Orfanidis, S.J.: Introduction to Signal Processing. Prentice-Hall (1995)
3. Ding, H., Lu, J., Qiu, X., Xu, B.: Anadaptive speech enhancement method for siren noise cancellation. Applied Acoustics 65, 385–399 (2004)
4. Michalewicz, Z.: Genetic algorithms + data structures = evolution programs. Springer, Heidelberg (1992)
5. Takaaki, N., Takahiko, K., Keiichiro, Y.: Deterministic Genetic Algorithm. Papers of Technical Meeting on Industrial Instrumentation and Control, IEE Japan, pp. 33-36 (2003)
6. Słowik, A., Białko, M.: Modified Version of Roulette Selection for Evolution Algorithms – The Fan Selection. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 474–479. Springer, Heidelberg (2004)
7. Słowik, A.: Steering of Balance between Exploration and Exploitation Properties of Evolutionary Algorithms - Mix Selection. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS (LNAI), vol. 6114, pp. 213–220. Springer, Heidelberg (2010)
8. Dorigo, M., Maniezzo, V., Colorni, A.: Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on SMC-B 26(1), 29–41 (1996)

9. Bilchev, G., Parmee, I.C.: The ant colony metaphor for searching continous design spaces. In: Fogarty, T.C. (ed.) AISB-WS 1995. LNCS, vol. 993, pp. 25–39. Springer, Heidelberg (1995)
10. Eberhart, R.C.: Kennedy J.A.: New optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, pp. 39–43 (1995)
11. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)
12. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
13. Slowik, A.: Hybridization of Evolutionary Algorithm with Yule Walker Method to Design Minimal Phase Digital Filters with Arbitrary Amplitude Characteristics. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS (LNAI), vol. 6678, pp. 67–74. Springer, Heidelberg (2011)
14. Karaboga, N., Cetinkaya, B.: Design of minimum phase digital IIR filters by using genetic algorithm. In: Proc. 6th Nordic Signal Processing Symposium 1, CD-ROM (2004)
15. Nakamoto, M., Yoshiya, T., Hinamoto, T.: Finite wordlength design for IIR digital filters based on the modified least-square criterion in the frequency domain. In: Int. Symposium on Intelligent Signal Processing and Communication Systems, ISPACS, vol. 1, pp. 462–465 (2007)
16. Baicher, G.S.: Optimization of finite word length coefficient IIR digital filters through genetic algorithms a comparative study. In: Jiao, L., Wang, L., Gao, X.-b., Liu, J., Wu, F. (eds.) ICNC 2006. LNCS, vol. 4222, pp. 641–650. Springer, Heidelberg (2006)
17. Karaboga, N., Cetinkaya, B.: Performance comparison of genetic algorithm based design methods of digital filters with optimal magnitude response and minimum phase. In: 46th IEEE Midwest Symposium on Circuits and Systems 1, CD-ROM (2003)
18. Moraglio, A., Togelius, J.: Geometric differential evolution. In: Proceeding GECCO 2009 Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 1705–1712 (2009)
19. Storn, R., Price, K.V.: Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. Global Optimization 11, 341–359 (1997)
20. Price, K.V.: An Introduction to Differential Evolution. New Ideas in Optimization, 79–108 (1999)
21. Lyons, R.: Introduction to Digital Signal Processing. WKL, Warsaw (2000) (in polish)
22. Bocewicz, G., Wójcik, R., Banaszak, Z.: Agvs distributed control subject to imprecise operation times. In: Nguyen, N.T., Jo, G.-S., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2008. LNCS (LNAI), vol. 4953, pp. 421–430. Springer, Heidelberg (2008)
23. Bocewicz, G., Banaszak, Z.: Declarative modeling of multimodal cyclic processes. In: Golinska, P., Fertsch, M., Marx-Gomez, J. (eds.) Information Technologies in Environmental Engineering. Environmental Science and Engineering - Environmental Engineering, vol. 3, pp. 551–568. Springer, Heidelberg (2011)

Bio-inspired Optimization of Thermomechanical Structures

Mirosław Szczepanik[1,*], Arkadiusz Poteralski[1], Adam Długosz[1], Wacław Kuś[1], and Tadeusz Burczyński[1,2]

[1] Silesian University of Technology, Gliwice, Poland
{miroslaw.szczepanik,arkadiusz.poteralski,
adam.dlugosz,waclaw.kus}@polsl.pl
[2] Cracow University of Technology, Cracow, Poland
tburczyn@pk.edu.pl

Abstract. The paper is devoted to an application of particle swarm optimizer and artificial immune system to optimization of elastic bodies under thermomechanical loading. The optimization problem is formulated as minimization of the volume, the maximal value of the equivalent stress, the maximal value of the temperature or maximization of the total dissipated heat flux with respect to specific dimensions of a structure. The direct problem is computed by means of the finite element method. Numerical examples for shape optimization are also included.

Keywords: particle swarm optimizer (PSO), artificial immune system (AIS), optimization, finite element method (FEM), thermoelasticity, computational intelligence.

1 Introduction

Optimal properties of structures can be searched using the computer aided optimization techniques. An appropriate operation of structures can be established by changing their shape, topology and material properties. The choice of optimal shape and topology of the structure decides about the effectiveness of the construction.

Shape and topology optimization of the structures under thermomechanical loadings are an active research area [1 – 3]. The paper deals with an application of particle swarm optimizer or artificial immune system and the finite element method to the optimization problems of a heat radiators used to dissipate heat from electrical devices. The main feature of the bio-inspired methods is to simulate biological processes. AIS is based on the mechanism discovered in biological immune systems. The main purpose of the immune system is to recognize and destroy pathogens - funguses, viruses, bacteria and improper functioning cells. The optimization process using AIS is based on finding the most dangerous pathogen as the global optimum of the objective function. PSO is based on the models of the animals social behaviours: moving and

* Corresponding author.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 79–90, 2013.

living in the groups. PSO algorithm realizes directed motion of the particles in n-dimensional space to search for solution for n-variable optimization problem. The optimization process using PSO is based on finding the better and better locations in the search-space (in the natural environment that are for example hatching or feeding grounds). The main advantage of the bio-inspired methods is the fact that these approachs do not need any information about the gradient of the fitness function and give a strong probability of finding the global optimum. The main drawback of these approaches is the long time of calculations. The fitness function is calculated for each swarm particle in each iteration by solving the boundary-value problem by means of the finite element method (FEM).

2 The Intelligent Bio-inspired Optimization Methods

The main advantage of the intelligent bio-inspired optimization methods like: the evolutionary algorithm [4-10], the artificial immune system [11-13] and the particle swarm optimizer [14-16] is the fact that these approaches do not need any information about the gradient of the fitness function and give a strong probability of finding the global optimum. Two different bio-inspired optimization methods, like the artificial immune system and the particle swarm optimizer are applied in this work.

2.1 Artificial Immune System

The artificial immune systems [11] are developed on the basis of a mechanism discovered in biological immune systems. The cloning algorithm Clonalg presented by von Zuben and de Castro [17] uses some mechanisms similar to biological immune systems to global optimisation problems. The unknown global optimum is the searched pathogen. The memory cells contain design variables and proliferate during the optimisation process. The B cells created from memory cells undergo mutation. The B cells evaluate and better ones exchange memory cells. In Wierzchoń version of Clonalg [11] the crowding mechanism is used - the diverse between memory cells is forced. A new memory cell is randomly created and substitutes the old one, if two memory cells have similar design variables. The crowding mechanism allows finding not only the global optimum but also other local ones. The presented approach is based on the Wierzchoń algorithm [11], but the mutation operator is changed. The Gaussian mutation is used instead of the nonuniform mutation in the presented approach. Figure 1. presents the flowchart of an artificial immune system. The memory cells are created randomly. They proliferate and mutate creating B cells. The number of clones created by each memory cell is determined by the memory cells objective function value. The objective functions for B cells are evaluated. The selection process exchanges some memory cells for better B cells. The selection is performed on the basis of the geometrical distance between each memory cell and B cells (measured by using design variables). The crowding mechanism removes similar memory cells. The similarity is also determined as the geometrical distance between memory cells.

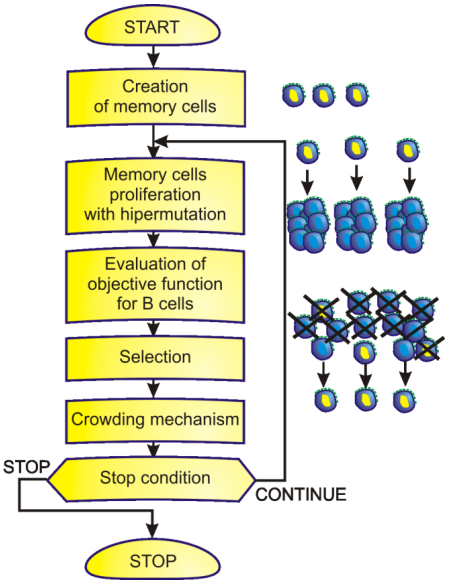


Fig. 1. An artificial immune system

The process is iteratively repeated until the stop condition is fulfilled. The stop condition can be expressed as the maximum number of iterations.

2.2 Particle Swarm Optimizer

The particle swarm algorithms [14], similarly to the evolutionary and immune algorithms, are developed on the basis of the mechanisms discovered in the nature. The swarm algorithms are based on the models of the animals social behaviours: moving and living in the groups. The Particle Swarm Optimizer – PSO has been proposed by Kennedy and Eberhart [14]. This algorithm realizes directed motion of the particles in n-dimensional space to search for solution for n-variable optimization problem. PSO works in an iterative way.

The algorithm with continuous representation of design variables and constant constriction coefficient (constricted continuous PSO) has been used in presented research. In this approach each particle oscillates in the search space between its previous best position and the best position of its neighbours, hopefully finding new best locations on its trajectory. When the swarm is rather small (swarm consists of several or tens particles) it can be assumed that all particles are the neighbours of currently considered one. In this case we can assume the global neighbourhood version and the best location found by swarm so far is taken into account – current position of the swarm leader (Figure 2a).

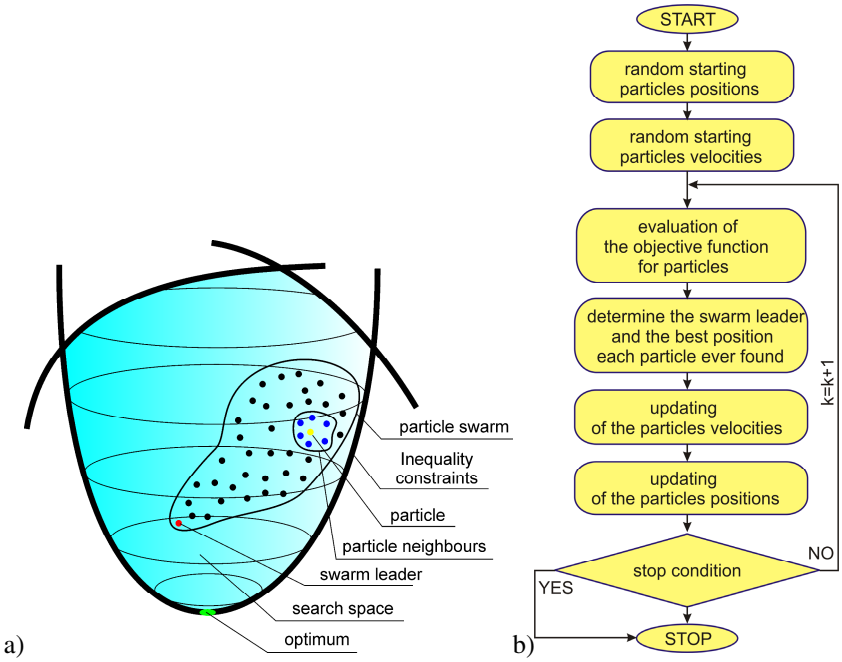


Fig. 2. a) The idea of the particle swarm, b) Particle swarm optimizer - block diagram

The position x_{ij} of the *i-th* particle is changed by stochastic velocity v_{ij}, which is dependent on the particle distance from its earlier best position and position of the swarm leader. This approach is given by the following equations:

$$v_{ij}(k+1) = wv_{ij}(k) + \phi_{1j}(k)\left[q_{ij}(k) - d_{ij}(k)\right] + \phi_{2j}(k)\left[\hat{q}_j(k) - d_{ij}(k)\right] \tag{1}$$

$$d_{ij}(k+1) = d_{ij}(k) + v_{ij}(k+1), \quad i = 1,2,...,m \; ; \; j = 1,2,...,n \tag{2}$$

where: $\phi_{1j}(k) = c_1 r_{1j}(k)$; $\phi_{2j}(k) = c_2 r_{2j}(k)$, m – number of the particles, n – number of design variables (problem dimension), w – inertia weight, c_1, c_2 – acceleration coefficients, r_1, r_2 – random numbers with uniform distribution [0,1], $d_i(k)$ – position of the i-th particle in k-th iteration step, $v_i(k)$ – velocity of the i-th particle in k-th iteration step, $q_i(k)$ – the best found position of the i-th particle found so far, $\hat{q}(k)$ – the best position found so far by swarm – the position of the swarm leader, k – iteration step.

The flowchart of the particle swarm optimizer is presented in Figure 2b. At the beginning of the algorithm the particle swarm of assumed size is created randomly. The objective function values are evaluated for each particle. In the next step the best positions of the particles are updated and the swarm leader is chosen. Then the particle velocities are modified by means of the Equation (1) and particle positions are modified according to the Equation (2). The process is iteratively repeated until the stop condition is fulfilled.

3 Evaluation of the Fitness Function

The fitness function is computed with the use of the steady-state thermoelsticity. Elastic body occupied the domain Ω bounded by the boundary Γ is considered (Figure 3).

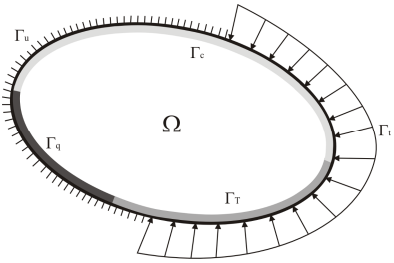


Fig. 3. Elastic structure subjected to thermomechanical boundary conditions

The governing equations of the linear elasticity and steady-state heat conduction problem is expressed by the following equations:

$$G\,u_{i,jj} + \frac{G}{1-2\nu}u_{j,ji} + \frac{2G(1-\nu)}{1-2\nu}\alpha T_{,i} = 0 \tag{3}$$

$$\lambda T_{,ii} + Q = 0 \tag{4}$$

where G is a shear modulus and ν is a Poisson ratio, u_i is a field of displacements, α is heat conduction coefficient, λ is a thermal conductivity, T is a temperature and Q is an internal heat source.

The mechanical and thermal boundary conditions for the equations (3) and (4) take the form:

$$\Gamma_t : t_i = \bar{t}_i \;;\; \Gamma_u : u_i = \bar{u}_i \;;\; \Gamma_T : T_i = \bar{T}_i \;;\; \Gamma_q : q_i = \bar{q}_i \;;\; \Gamma_c : q_i = \alpha(T_i - T^{\infty}) \tag{5}$$

where $\bar{u}_i, \bar{t}_i, \bar{T}_i, \bar{q}_i, \alpha, T^{\infty}$ is known displacements, tractions, temperatures, heat fluxes heat conduction coefficient and ambient temperature respectively. Separate parts of the boundaries must fulfill the following relations:

$$\Gamma = \Gamma_t \cup \Gamma_u = \Gamma_T \cup \Gamma_q \cup \Gamma_c; \quad \Gamma_t \cap \Gamma_u = \varnothing; \quad \Gamma_T \cap \Gamma_q \cap \Gamma_c = \varnothing \tag{6}$$

In order to solve numerically thermoelasticity problem finite element method is proposed. After discretization taking into account boundary conditions following system of linear equations can be obtained:

$$\begin{aligned} \mathbf{KU} &= \mathbf{F} \\ \mathbf{ST} &= \mathbf{R} \end{aligned} \tag{7}$$

where **K** denotes stiffness matrix, **S** denotes conductivity matrix, **U**, **F**, **T**, **R** contain discretized values of the boundary displacements, forces, temperatures and heat

fluxes. This problem is solved by the FEM software – MENTAT/MARC [18]. The preprocessor MENTAT enables the production of the geometry, mesh, material properties and settings of the analysis. In order to evaluate the fitness function for each particle following four steps must be performed:

Step 1 (*generated using MENTAT*) Create geometry and mesh on the base of the particles; **Step 2** *(generated using MENTAT)* Create the boundary conditions, material properties, settings of the analysis; **Step 3** *(solved using MARC)* Solve thermoelasticity problem; **Step4** Calculate the fitness functions values on the base of the output MARC file

4 Formulation of the Optimization Problem

The problem of the optimal shape of a heat radiator used to dissipate heat from electrical devices is considered [19]. The exemplary heat exchangers are presented in Fig. 4.

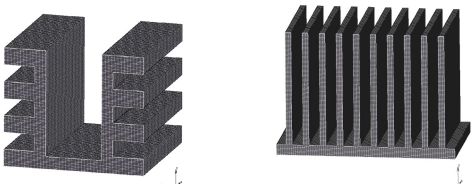


Fig. 4. Proposed geometry of considered heat radiators

The shape optimization problem is solved by the minimization of appropriate functionals. In the present paper following functionals are proposed:

• The volume of the structure defined as:

$$\min_{\mathrm{X}} V(\mathrm{X}) \tag{8}$$

with imposed constrains on the maximal value of temperature ($T - T^{ad} \le 0$) and the maximal value of equivalent stress ($\sigma_{eq} - \sigma_{eq}^{ad} \le 0$).

• The minimization of the maximal value of the equivalent stress defined as:

$$\min_{\mathrm{X}} \sigma_{eq}^{\max}(\mathrm{X}) \tag{9}$$

• The minimization of the maximal value of the temperature in the structure defined as:

$$\min_{\mathrm{X}} T^{\max}(\mathrm{X}) \tag{10}$$

with imposed constrains on the maximal value of volume of the structure ($V - V^{ad} \le 0$).

X is the vector of design parameters which is represented by a particle with the floating point representation. The heat radiator is modelled as a two dimensional (2D) plain stress problem. The fitness function is computed with the use of the steady-state

thermoelsticity. The governing equations of the linear elasticity and steady-state heat conduction problem are expressed by the following equations:

$$G\,u_{i,jj} + \frac{G}{1-2v}u_{j,ji} + \frac{2G(1-v)}{1-2v}\alpha T_{,i} = 0 \tag{11}$$

$$\lambda T_{,ii} + Q = 0 \tag{12}$$

where G is a shear modulus and v is a Poisson ratio, u_i is a field of displacements, α is heat conduction coefficient, λ is a thermal conductivity, T is a temperature and Q is an internal heat source.

The mechanical and thermal boundary conditions for the equations (11) and (12) take the form:

$$\Gamma_t : t_i = \bar{t}_i \;;\; \Gamma_u : u_i = \bar{u}_i \;;\; \Gamma_T : T_i = \bar{T}_i \;;\; \Gamma_q : q_i = \bar{q}_i \;;\; \Gamma_c : q_i = \alpha(T_i - T^{\infty}) \tag{13}$$

where $\bar{u}_i, \bar{t}_i, \bar{T}_i, \bar{q}_i, \alpha, T^{\infty}$ is known displacements, tractions, temperatures, heat fluxes heat conduction coefficient and ambient temperature respectively.

In order to solve numerically thermoelasticity problem finite element method (FEM) is used [20]. After discretization taking into account boundary conditions the following system of linear equations can be obtained:

$$\mathbf{KU = F} \qquad \mathbf{ST = R} \tag{14}$$

where **K** denotes stiffness matrix, **S** denotes conductivity matrix, **U, F, T, R** contain discretized values of the boundary displacements, forces, temperatures and heat fluxes. The commercial FEM software – Mentat/Marc [19] is used.

5 Geometry Modeling

The choice of the geometry modeling method and the design variables has a great influence on the final solution of the optimization process. There is a lot of methods for geometry modeling. In the proposed approach Bezier curves are used to model the geometry of the structures. This type of the curve is a superset of the more commonly known NURBS (Non-Uniform Rational B-Spline). Using these curves in optimization makes the reduction of the number of design parameters possible. By manipulating the control points it provides the flexibility to design a large variety of shapes.

An nth-degree Bezier curve is defined by:

$$C(u) = \sum_{i=0}^{n} B_{i,n}(u) P_i \tag{15}$$

where u is a coordinate with changes range <0,1>, P_i are control points.

The basis functions $B_{i,n}$ are given by:

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!}u^i (1-u)^{n-1} \tag{16}$$

The 4-th degree Bezier curve is described by the following equation:

$$C(u) = (1-u)^4 P_0 + 4u(1-u)^3 P_1 + 6u^2(1-u)^2 P_2 + 4u^3(1-u)P_3 + u^4 P_4 \tag{17}$$

An example of the 4-th Bezier curves is shown in Figure 5. By manipulating the control points, it provides the flexibility to design a large variety of shapes.

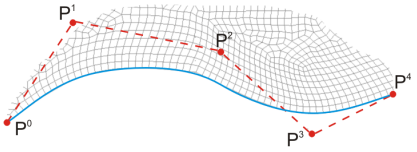


Fig. 5. The example modeling of the shape of the structure by 4th-degree Bezier curve

By changing the value of *u* between *0* and *1* successive points of the curve are obtained. For *u=0* $C(u)=P^0$ and for *u=1* $C(u)=P^4$. The shapes of Bezier curve depend on the position of control points. In order to obtain more complicated shapes, it is necessary to raise up the degree of the Bezier curve and introduce more control points.

6 Numerical examples

6.1 Example 1

The shape optimization problem is solved by the minimization of the volume of the structure with constrains imposed on the temperature and equivalent stress ($\sigma_{eq}^{ad} = 40MPa$). Three cases of constraints of the temperature were considered ($T^{ad} = 90,\ 100,\ 110\,°C$). Geometry, scheme of loading and the distribution of 5 design parameters are presented in Fig. 6. Parameters of particle swarm optimizer and artificial immune system are included in Tab.1. Boundary conditions values are presented in Tab. 2. Tab. 3 includes the admissible values of the design parameters and results of optimization.

Table 1. Parameters of PSO and boundary conditions values

Parameters of PSO		Parameters of AIS	
Number of particles	15	Number of memory cells	5
Inertia weight *w*	0.73	Number of the clones	5
Acceleration coefficient c_1	1.47	Crowding factor	0.45
Acceleration coefficient c_2	1.47	Gaussian mutation	50%

Table 2. Boundary condition values

Boundary condition values			
Dissipated heat	P	Ambient temperature	Heat convection coefficient
80W	10N	25 °C	2W/m^{2}K

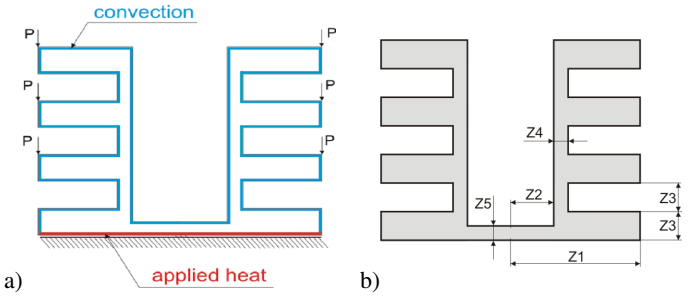


Fig. 6. Geometry and scheme of loading b) Design parameters

Table 3. The admissible values of the design parameters and results of optimization

The admissible values of the design parameters							
Design variable		Z1 [mm]	Z2 [mm]	Z3 [mm]	Z4 [mm]	Z5 [mm]	
Range		20÷100	2÷10	4÷10	4÷10	4÷10	
Results of optimization using PSO							
	Z1 [mm]	Z2 [mm]	Z3 [mm]	Z4 [mm]	Z5 [mm]	Volume [mm^3]	F.f.eval
T^{ad}=90°C	43.62	2.86	4.00	4.29	4.00	22884	1860
T^{ad}=100°C	39.42	4.99	4.00	4.00	4.81	19935	135
T^{ad}=110°C	32.09	2.00	4.00	4.00	4.00	17199	480
Results of optimization using AIS							
	Z1 [mm]	Z2 [mm]	Z3 [mm]	Z4 [mm]	Z5 [mm]	Volume [mm^3]	F.f.eval
T^{ad}=90°C	47.87	7.83	4.00	4.17	4.00	23102	1068
T^{ad}=100°C	44.80	10.00	4.00	4.00	4.00	20631	552
T^{ad}=110°C	34.69	5.03	4.00	4.00	4.00	17361	912

6.2 Example 2

The shape optimization problem modelled using Bezier curve is solved by the minimization of the three fitness functions: volume of the structure with constrains imposed on the temperature and equivalent stress ($\sigma_{eq}^{ad} = 15MPa$), temperature and equivalent stresses with constraints imposed on the volume of the structures. Geometry, scheme of loading and the distribution of design parameters are presented in Fig. 7. Parameters of PSO and AIS and boundary conditions values are presented in Tab. 1 and 4, respectively. Tab. 5 includes the admissible values of the design parameters and Tab. 6 - results of optimization.

Table 4. Boundary condition values

Boundary condition values			
Pressure	Heat flux	Ambient temperature	Heat convection coefficient
5000Pa	1000W/m^2	25 °C	2W/m^{2}K

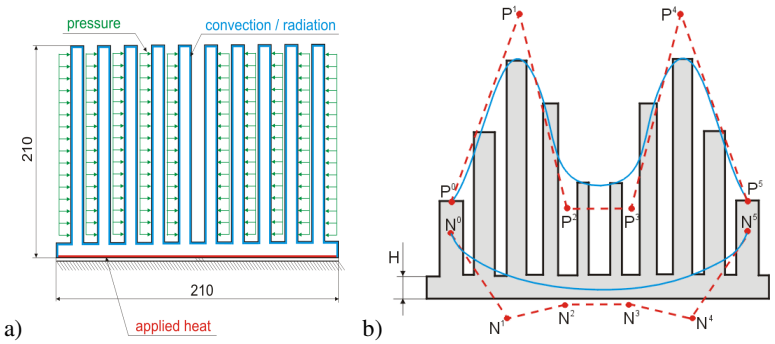


Fig. 7. Geometry and scheme of loading b) Design parameters

Table 5. The admissible values of the design parameters

Design variable	P0 [mm]	P1 [mm]	P2 [mm]	P3 [mm]	P4 [mm]	P5 [mm]	H [mm]
Range	30÷200	30÷200	30÷200	30÷200	30÷200	30÷200	7÷15
Design variable	N0 [mm]	N1 [mm]	N2 [mm]	N3 [mm]	N4 [mm]	N5 [mm]	
Range	4÷12	4÷12	4÷12	4÷12	4÷12	4÷12	

Table 6. Results of optimization

P0= P5	P1= P4	P2= P3	N0= N5	N1= N4	N2= N3	H
PSO						
Results of optimization (minimization of temperature)						
174.1	200	104.5	4	4	4	7
Fitness function evaluation					58.22°C	
Results of optimization (minimization of volume)						
83.9	45.8	73.7	4	4	4	7
Fitness function evaluation					0.0007719 m3	
Results of optimization (minimization of equivalent stresses)						
30	30	30	12	11.98	11.94	8.19
Fitness function evaluation					0.13 MPa	
AIS						
Results of optimization (minimization of temperature)						
186.3	144	141	4	4	4.01	7
Fitness function evaluation					58.29°C	
Results of optimization (minimization of volume)						
95.5	75	30	4	4	4	7
Fitness function evaluation					0.0007731 m3	
Results of optimization (minimization of equivalent stresses)						
30	30	35.1	9.01	10.03	11.1	9.42
Fitness function evaluation					0.15 MPa	

7 Conclusions

An effective tools of bio-inspired optimization of elastic bodies under thermomechanical loading are presented. Using this approaches the optimal shape of a heat radiators used to dissipate heat from electrical devices is obtained. Described approaches are free from limitations connected with classic gradient optimization methods referring to the continuity of the objective function, the gradient or hessian of the objective function and the substantial probability of getting a local optimum. Comparison between PSO and AIS shows that PSO is more effective than AIS in considered optimization problem. The results of the numerical examples confirm the efficiency of the proposed optimization methods and demonstrate that the method based on soft computing is an effective technique for solving computer aided optimal design problems.

Acknowledgment. The research is partially financed as a research project no. BK-204/RMT4/2012.

References

1. Hansen, G., Bendsøe, M.P., Sigmund, O.: Topology optimization of heat conduction problems using the finite volume method. Struct. Multidisciplin. O. 31(4), 251–259 (2006)
2. Cho, S., Choi, J.Y.: Efficient topology optimization of thermo-elasticity problems using coupled field adjoint sensitivity analysis method. FEA, Des. 41(15), 1 481-1 495 (2005)
3. Li, Q., Steven, G.P., Xie, Y.M., et al.: Evolutionary topology optimization for temperature reduction of heat conducting fields. Int. J. Heat Mass Transfer 47(23), 5 071–5 083 (2004)
4. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolutionary Programs. Springer, Berlin (1992)
5. Burczyński, T., Poteralski, A., Szczepanik, M.: Topological evolutionary computing in the optimal design of 2D and 3D structures. Eng. Opt. Taylor & Francis 39(7), 811–830 (2007)
6. Burczyński, T., Kuś, W., Długosz, A., Poteralski, A., Szczepanik, M.: Sequential and Distributed Evolutionary Computations in Structural Optimization. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 1069–1074. Springer, Heidelberg (2004)
7. Burczyński, T., Poteralski, A., Szczepanik, M.: Genetic generation of 2-D and 3-D structures Second M.I.T. Conference on Computational Fluid and Solid Mechanics Massachusetts, Institute of Technology Cambridge, MA 02139 U.S.A (2003)
8. Burczynski, T., Dlugosz, A., Kus, W., Orantek, P., Poteralski, A., Szczepanik, M.: Intelligent computing in evolutionary optimal shaping of solids. In: Proc. in. 3rd International Conference on Computing, Communications and Control Technologies, vol. 3, pp. 294–298 (2005)
9. Długosz, A.: Evolutionary computation in thermoelastic problems. In: IUTAM Symposium on Evolutionary Methods in Mechanics, vol. 117, pp. 69–80 (2004)
10. Długosz, A., Burczynski, T.: Multiobjective shape optimization of selected coupled problems by means of evolutionary algorithms. Bulletin of the Polish Academy of Sciences, Technical Sciences 60(2), 215–222 (2012)
11. Wierzchoń, S.: Artificial Immune Systems, theory and applications. EXIT (2001)

12. Burczyński, T., Bereta, M., Poteralski, A., Szczepanik, M.: Immune Computing: Intelligent Methodology and its applications in bioengineering and computational mechanics.In: Advanced Structured Materials v1. Comput. Meth. Mech. Springer (2010)
13. Poteralski, A., Szczepanik, M., Dziatkiewicz, G., Kuś, W., Burczyński, T.: Immune identification of piezoelectric material constants using BEM. Inverse Probl. Sci. Eng. (2011)
14. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann (2001)
15. Szczepanik, M., Burczyński, T.: Swarm optimization of stiffeners locations in 2-D structures. Bulletin of the Polish Academy of Sciences, Technical Sciences 60(2), 241–246 (2012)
16. Szczepanik, M., Poteralski, A., Ptaszny, J., Burczyński, T.: Hybrid Particle Swarm Optimizer and Its Application in Identification of Room Acoustic Properties. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) EC 2012 and SIDE 2012. LNCS, vol. 7269, pp. 386–394. Springer, Heidelberg (2012) ISBN 978-3-642-29352-8
17. de Castro, L.N., Von Zuben, F.J.: Immune and neural network models: theoretical and empirical comparisons. IJCIA 1(3), 239–257 (2001)
18. MSC.MARC Theory and user information, vol. A-D. MSC Software Corporation (2007)
19. Burczyński, T., Długosz, A., Kuś, W.: Parallel Evolutionary Algorithms in Shape Optimization of Heat Radiators. JTAM 44(2), 351–366 (2006)
20. Zienkiewicz, O.C., Taylor, R.L.: The Finite Element Method. Butterworth Heinemann, Oxford (2000)

Some Aspects of Evolutionary Designing Optimal Controllers

Jacek Szczypta, Andrzej Przybył, and Krzysztof Cpałka

Częstochowa University of Technology,
Institute of Computational Intelligence, Poland
{jacek.szczypta,andrzej.przybyl,krzysztof.cpalka}@iisi.pcz.pl

Abstract. In this paper a new automatic method of control system design was presented. Our method is based on the evolutionary algorithm, which is used for selection of the controller structure as well as for parameters tuning. This is realized by means of testing different controller structures and elimination of spare elements, taking into account theirs impact on control quality factors. Presented method was tested with two control objects of different complexity.

1 Introduction

Automatic control is an important issue from scientific and practical point of view (see e.g. [24]). It has a significant impact on the quality and efficiency of industrial processes and human safety. It should be noted that in many practical cases the automatic control systems do not have the optimal structure or parameters. This is due to difficult and time-consuming process of selecting the optimal structure and parameters of the actual control system. Commonly used design process of control systems usually relies on the knowledge and experience of experts. Design process also uses simplified (i.e. usually linear) model of controlled objects. The selection of the optimal control system for real control object (different from the simplified model) must be carried out by trial and error. In general, every admissible control structure should be tested and on this basis chosen is the best one.

In the scientific literature much attention was devoted to the controller design issues. There are, among other ideas, described model-based controllers - i.e. which need the model and parameters of the controlled object (for example the model reference adaptive controllers (MRAC) see e.g. [25]), controllers which do not need the object model - like PID controllers (see e.g. [19], [20]) and sliding mode controllers (see e.g. [5]). In the last years there were widely used a nonlinear controllers which are based on soft-computing techniques (see e.g. [10], [12], [17]). Some authors use soft-computing techniques in synthesis of controllers structure (see e.g. [14]).

The controller that best meets the specific requirements of a given control system is chosen from a set of various controller structures of different properties. The choice is done by the human expert, based on his knowledge and experience.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 91–100, 2013.

In comparison with other methods available in the literature, the method presented in this paper allows for both controller structure selection and parameter tuning. This is done automatically using an evolutionary algorithm and based on the accurate model of the controlled object and the controller (see e.g. [1]). As a result, the optimal, i.e. maximizing of the value of used performance index, controller is designed.

This paper is organized into five sections. In Section 2 we describe an idea of the new method for designing optimal controllers. In Section 3 we present a detailed description of the new method for designing optimal controllers. In Section 4 simulation results are presented. Conclusions are drawn in Section 5.

2 Idea of the New Method for Designing Optimal Controllers

The method presented in this paper is dedicated for both controller structure selection and parameter tuning (Fig. 1). This is realized automatically using evolutionary algorithm and based on the as far as possible accurate controller and model of the controlled object. The idea of the method is that the evolutionary algorithm uses the simulation model with the constraints and nonlinearities of the actuators, the controlled object, measurement errors etc. As a result of the evolutionary algorithm activity, the controller is well suited to work with real conditions. This is not possible with the use of any other (i.e. analytical) tuning methods (see e.g. [18], [22], [23]).

The important part of the presented method is the ability to select the optimal controller structure. The optimal structure selection occurs, because during evolutionary learning some controller parts may be eliminated (see e.g. [2], [4]). Of course, elimination does not have a significant negative impact on control quality.

The aim of the evolutionary algorithm is to maximize properly defined fitness function. Its value depends on a few control quality indicators such as: root mean square error (RMSE) value (i.e. difference between required and current control signals), controller structure complexity, existence of the overshoot of the control signals, total harmonic distortion (THD) of the control signals etc. Thus, the obtained controller is optimal, i.e. it maximizes the value of used performance index. Fitness function evaluation involves testing of controller, with its complex structure and appropriate parameters, on a realistic model of the whole control system. Fitness function will be described in the next part of this paper.

Steps of the method used in this paper are the same as in typical evolutionary algorithm (see e.g. [3], [6] - [9], [11], [13], [15], [21]), and are as follows:

- **Step 1.** Initialize chromosome population. Every chromosome codes a single structure of the controller and its parameters.
- **Step 2.** Chromosome population evaluation. Aim of this step is evaluation of the control systems coded in the population of the chromosomes in the sense of adopted criterion.

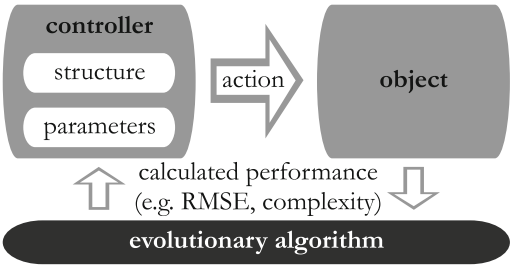


Fig. 1. Idea of the new method for optimal control structure and parameter set selection

- **Step 3.** Evolutionary algorithm stop condition checking. When the best chromosome in population (coding information about best control system in the sense of adopted control criterion) satisfies the stop condition, the algorithm returns information about this chromosome and exits. Otherwise, the algorithm goes to step 4.
- **Step 4.** Chromosome selection for evolutionary operations.
- **Step 5.** Crossover and mutation operators application. This step includes also repair of chromosomes that were obtained from evolutionary operations. Aim of the repair is to correct values of the genes, that are coding control system parameters, to preserve in acceptable range.
- **Step 6.** Generate offspring chromosome population and then go to step 2.

In section 3 we present a detailed description of the new method for designing optimal controllers.

3 Detailed Description of the New Method for Designing Optimal Controllers

Detailed information aboutthe proposed in this paper method of optimal controller design can be summarized as follows:

- **Initialization.** Parent chromosome population initialization includes random generation of genes values. Values are taken from search range (see e.g. [7], [8]). Search range should be customized individually for every single problem.
- **Coding.** Every chromosome $\mathbf{X}_{ch}^{\text{par}}$ codes full parameter set of control system. In this paper the controller composed of typical PID controllers (see e.g. [16]) is considered (Fig. 2). Every gene of the chromosome codes single real value of some controller parameter. Chromosome $\mathbf{X}_{ch}^{\text{par}}$ is described as follows:

$$\mathbf{X}_{ch}^{\text{par}} = (P_1, I_1, D_1, P_2, I_2, D_2, \ldots) = \left(X_{ch,1}^{\text{par}}, X_{ch,2}^{\text{par}}, \ldots, X_{ch,L}^{\text{par}}\right), \quad (1)$$

where P_1, I_1, D_1,..., denote control system parameter values, $ch = 1, .., Ch$, Ch denotes a number of chromosomes in the population, L denotes length of

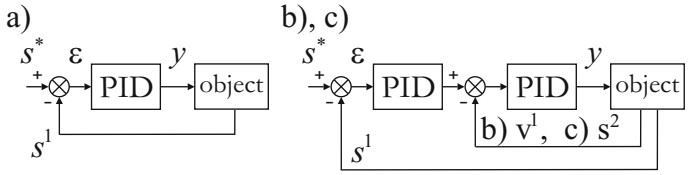


Fig. 2. Control systems structures tested in the experiments: a) 1xPID-s b) 2xPID-v c) 2xPID-s

the chromosome $\mathbf{X}_{ch}^{\text{par}}$. It is important to notice that, for each chromosome $\mathbf{X}_{ch}^{\text{par}}$ in population there is chromosome $\mathbf{X}_{ch}^{\text{red}}$. Binary genes of the chromosome $\mathbf{X}_{ch}^{\text{red}}$ decide which parts of elementary PID controllers should occur in control system. Chromosome $\mathbf{X}_{ch}^{\text{red}}$ is described as follows:

$$\mathbf{X}_{ch}^{\text{red}} = \left(X_{ch,1}^{\text{red}}, X_{ch,2}^{\text{red}}, \ldots, X_{ch,L}^{\text{red}}\right), \tag{2}$$

where every gene in $X_{ch,g}^{\text{red}} \in \{0,1\}$, $ch = 1,..,Ch$, $g = 1,..,L$, decides, if relevant part of control system occurs in control process (relevant gene $X_{ch,g}^{\text{red}} = 1$). It should be noted, that chromosome $\mathbf{X}_{ch}$, $ch = 1,..,Ch$, coding whole control system consists of two parts: part $\mathbf{X}_{ch}^{\text{par}}$ coding parameters and part $\mathbf{X}_{ch}^{\text{red}}$ coding structure.

- **Chromosome Selection.** Chromosome selection can be implemented using a method known from the literature (see e.g. [6], [21]). In our simulations, which results are presented in next section, roulette wheel selection was used. The idea of roulette wheel selection is to promote chromosomes with beneficial fitness function value.
- **Chromosome Crossover.** In this paper crossover with weighting of the genes values was assumed (see e.g. [6], [21]). In crossover take part only those chromosome pairs (selected in previous step), for which drawn real value from range $[0,1]$ is less than crossover probability p_c. $\mathbf{X}_{ch}^{\text{par}}$ parts crossover is described as follows:

$$\begin{cases} X_{j1,g}^{\text{par}} := (1-\phi)\cdot X_{j1,g}^{\text{par}} + \phi \cdot X_{j2,g}^{\text{par}} \\ X_{j2,g}^{\text{par}} := (1-\phi)\cdot X_{j2,g}^{\text{par}} + \phi \cdot X_{j1,g}^{\text{par}} \end{cases}, \tag{3}$$

where $j1$, $j2$ denote in-pair chromosome index, $g = 1,..,L$, denotes gene index, $\phi \in (0,1)$ denotes trial and error selected algorithm parameter. Of course for $\mathbf{X}_{ch}^{\text{par}}$ crossover can be used other method applicable for real coding. It is important to notice that $\mathbf{X}_{ch}^{\text{red}}$ crossover is the same as in classic genetic algorithm (see e.g. [15]).
- **Chromosome Mutation.** Only those chromosomes take part in mutation, for which drawn real value from range $[0,1]$ is less than mutation probability p_m. $\mathbf{X}_{ch}^{\text{par}}$ mutation is described as follows:

$$X_{ch,g}^{\text{par}} := X_{ch,g}^{\text{par}} + \sigma \cdot \text{random}\{-1,+1\} \cdot (pmax_g - pmin_g), \tag{4}$$

where $\sigma \in (0,1)$ denotes mutation intensity parameter, $pmax_g$, $g = 1,..,L$, denotes highest accepted gene value, $pmin_g$, $g = 1,..,L$, denotes lowest accepted gene value. It is important to notice that $\mathbf{X}_{ch}^{\text{red}}$ mutation is the same as in classic genetic algorithm (see e.g. [15].

– **Chromosome Repair.** The aim of the repair is to preserve in search range values of genes that are coding control system parameters. Chromosome repair is described as follows:

$$X_{ch,g}^{\text{par}} := \begin{cases} pmin_g & \text{for} \quad X_{ch,g}^{par} < pmin_g \\ pmax_g & \text{for} \quad X_{ch,g}^{par} > pmax_g \\ X_{ch,g}^{\text{par}} & \text{otherwise} \end{cases}. \tag{5}$$

– **Chromosome Evaluation.** Chromosome evaluation function was set to minimize: RMSE error, zero crossing number of controller output signal, controller output signal dynamics and overshoot of the control signal. High zero crossing number of controller output signal is a negative phenomenon, because it tends to excessive use of mechanical control parts and may cause often huge changes of the controller output signal value. While the overshoot of the control signal is not acceptable in many industrial applications (see e.g. [24]). Chromosome evaluation function is described as follows:

$$ff(\mathbf{X}_{ch}) = (RMSE_{ch} + c_{ch} \cdot w_c + z_{ch} \cdot w_z + ov_{ch} \cdot w_{ov})^{-1}, \tag{6}$$

where $c_{ch} > 0$ denotes the complexity of the controller structure and is calculated by the formula:

$$c_{ch} = \sum_{g=1}^{L} \mathbf{X}_{ch,g}^{\text{red}}, \tag{7}$$

$w_c \in [0,1]$ denotes weight factor for the complexity of the controller structure, $z_{ch} \geq 0$ denotes zero crossing number of controller output signal (in simulations its value is set automatically), $w_z \in [0,1]$ denotes weight for the zero crossing factor, $ov_{ch} \geq 0$ denotes value of the greatest overshoot of the controlled s^1 signal and finally $w_{ov} \in [0,1]$ denotes weight for the overshoot factor. RMSE error function of the ch chromosome is described by the following formula:

$$RMSE_{ch} = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} {\varepsilon_{ch,i}}^2}, \tag{8}$$

where $i = 1, \ldots, N$, denotes sample index, N denotes the number of samples, ε denotes controller tracking error defined as follows:

$$\varepsilon_{ch,i} = s_{ch}^{*} - s_{ch}^{1}, \tag{9}$$

where s^* denotes the value of the reference signal of the controlled value, while s^1 denotes its current value. In our method we maximize the function described in formula (6).

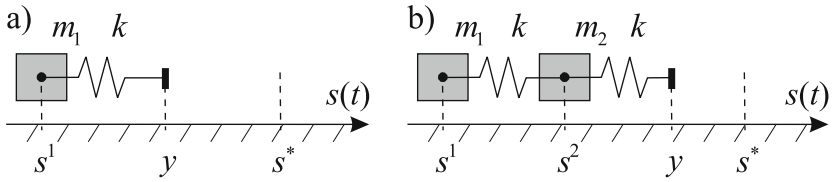


Fig. 3. Simulated spring-mass-damp objects: a) "type 1m", b) "type 2m"

4 Simulations Results

In our simulations there was considered a problem of design controller structure and parameter tuning for two cases: second and fourth-order controlled objects respectively:

- Single spring-mass-damp object, denoted as "type 1m" (Fig. 3.a).
- Double spring-mass-damp object, denoted as "type 2m" (Fig. 3.b).

We took the following assumptions in the simulations:

- Evolutionary algorithm population number was set to 100 chromosomes. Algorithm was executed for 300 iterations (generations), crossover probability value $p_c = 0.8$, mutation probability was $p_m = 0.2$, crossover weight ϕ was random value from range (0,1).
- Fitness function weights were set as follows $w_c = 0.01$, $w_z = 0.001$ and $w_{ov} = 0.001$.
- Simulation length was set to 10 seconds, a shape of the reference signal s^* is presented in Fig. 4 and Fig. 5.
- Search range for genes coding for controller parameter (5) were set as follows: $P_1 = [0,15]$, $I_1 = [0,15]$, $D_1 = [0,1]$, $P_2 = [0,15]$, $I_2 = [0,15]$ and $D_2 = [0,1]$.
- Output signal of the controller was limited to the range $y \in (-2, +2)$.
- Quantization resolution for the output signal y of the controller as well as for the position sensor for s^1 and s^2 was set to 10 bit.
- Time step in the simulation was equal to $T = 0.1$ms, while interval between subsequent controller activations were set to twenty simulation steps.
- The motion equations for position s^1, velocity v^1 and acceleration a^1 for object "type 1m" are described as follows:

$$s_n^1 = s_{n-1}^1 + v_{n-1}^1 \cdot T + (a_{n-1}^1 \cdot T^2) \cdot 0.5, \tag{10}$$

$$v_n^1 = v_{n-1}^1 + a_{n-1}^1 \cdot T, \tag{11}$$

$$a_n^1 = \left(\left(y_n - s_n^1\right) \cdot k - v_n^1 \cdot \mu\right) \cdot m_1^{-1}, \tag{12}$$

where n and $n-1$ denotes current and previous simulation step respectively, k is spring constant, y is controller output signal and μ is coefficient of kinetic friction. For object "type 2m" equations for position and velocity of mass

m_1 are identical as in object "type 1m", while the acceleration is described as follows:

$$a_n^1 = \left((s_n^2 - s_n^1) \cdot k - v_n^1 \cdot \mu\right) \cdot m_1^{-1}. \tag{13}$$

Analogically, for mass m_2, the motion equations for position s^2, velocity v^2 and acceleration a^2 have the following form:

$$s_n^2 = s_{n-1}^2 + v_{n-1}^2 \cdot T + (a_{n-1}^2 \cdot T^2) \cdot 0.5, \tag{14}$$

$$v_n^2 = v_{n-1}^2 + a_{n-1}^2 \cdot T, \tag{15}$$

$$a_n^2 = \left(\left(y - s_n^2\right) \cdot k - v_n^2 \cdot \mu\right) \cdot m_1^{-1}. \tag{16}$$

- Object parameters values were set as follows: spring constant k was set to 10 N/m, coefficient of friction $\mu = 0.5$, masses $m_1 = m_2 = 0.2$ kg. Initial values of: s^1, v^1, s^2 i v^2 were set to zero.
- During simulation three control systems were tested:
  - System with one PID controller (Fig. 2a) denoted as 1xPID-s.
  - Cascaded system combined from two PID controllers (Fig. 2b), where the internal controller was coupled with the signal v^1 denoted as 2xPID-v.
  - Cascaded system combined from two PID controllers (Fig. 2c), where the internal controller was coupled with the signal s^2 (denoted as 2xPID-s).

The idea of the controller structure optimization was to make test of the different controller structures in the same external conditions, calculate the quality factors and choose the best controller.

At the first stage there was investigated a control system whose task was to control the position of the spring-mass-dump object "type 1m" (Fig. 3a). The results are shown in Table 1 in columns named '1xPID+1m' and '2xPID-v+1m' and presented in Fig. 4b and Fig. 4c respectively. In order to better illustrate the control problem in Fig. 4a there was presented the behaviour of the open-loop control system. How we can see, the first controller structure (1xPID) was sufficient to control the second-order object. The use of more complex (cascaded) controller (2xPID-v) was not necessary. The RMSE values for both cases was similar.

In the next stage of the experiment the fourth-order control object "type 2m" was investigated. The results are shown in Table.1 in columns named '1xPID+2m', '2xPID-v+2m','2xPID-s+2m' and presented in Fig.5a-c respectively.

As we can see, the first controller structure (case 1xPID-s+2m) was unable to control this fourth-order object with an sufficient quality (Fig. 5a). The oscillation of the mass position was unacceptable. Similarly, the cascaded controller (2xPID-v) with an internal coupling signal v^1 did not provide a good control quality (Fig. 5b). Only the last controller structure was able to control this object with a good quality (Fig. 5c). The values of quality factors (i.e. RMSE values

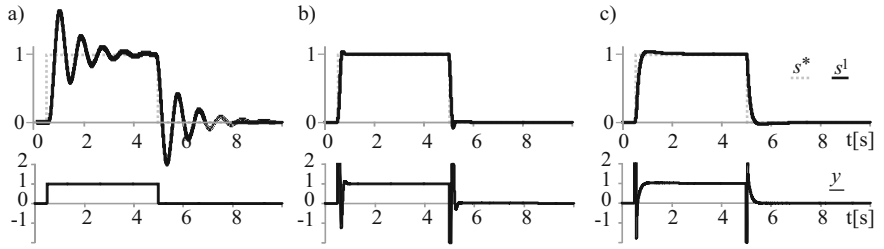


Fig. 4. Graph of the actual signal s^1 and reference value signal s^*, controller output signal y in simulation with the object "type 1m" and for controller: a) open-loop, b) 1xPID-s and c)2xPID-v

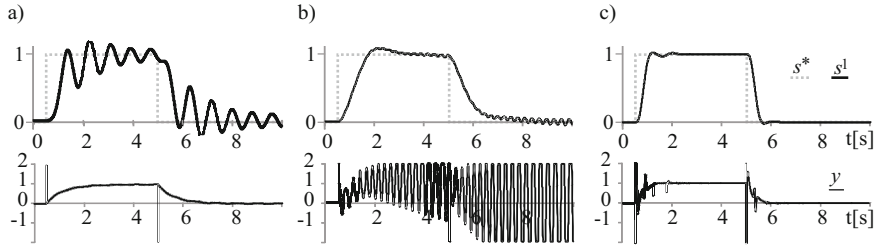


Fig. 5. Graph of the actual signal s^1 and reference value signal s^*, controller output signal y in simulation with the object "type 2m" and for controller: a) 1xPID-s, b) 2xPID-v and c) 2xPID-s

last row in Table 1) confirm the above observations. As a result, this quality factor together with properly defined fitness function (6) can be effectively used for the automatic optimization of the controller with the help of the evolutionary algorithm.

5 Summary

In this paper a new method for optimal controller design was proposed. A characteristic feature of this method is design of both controller parameter and structure using evolutionary algorithm. In order to prove the proposed method, three different controller structures with two different object types were tested. The simulations confirmed correctness of the proposed method.

Further research will be related to developing a general description of a variety of (admissible) control structures. This will allow automation of the design of the optimal controller structure to the specific control object, in the presence of various disturbances and other process control limitations.

Table 1. Parameters of evolutionary designed control systems

Name	Parameters of control systems				
	1xPID+1m	2xPID-v+1m	1xPID-s+2m	2xPID-v+2m	2xPID-s+2m
P_1	15.0	15.0	0.0217	2.06	1.01
I_1	9.63	reduced	1.37	3.08	6.03
D_1	0.7807	0.7850	0.03	0.15	0.20
P_2	-	6.139	-	0.50	1.07
I_2	-	15.0	-	0.08	0.04
D_2	-	reduced	-	0.20	1.60
$RMSE$	0.1209	0.1309	0.3344	0.3102	0.2296
$c_{ch} \cdot w_c$	0.03	0.04	0.03	0.06	0.06
$z_{ch} \cdot w_z$	0.008	0.008	0.008	0.063	0.021
$ov_{ch} \cdot w_o v$	0.0002	0.0001	0.0002	0.0001	0.0000
ff^-1	0.1591	0.179	0.3726	0.4333	0.3106
ff	6.287	5.586	2.684	2.308	3.219

Acknowledgment. The project was financed by the National Science Center on the basis of the decision number DEC-2012/05/B/ST7/02138.

References

1. Cordon, O., Herrera, F., Hoffman, F., Magdalena, L.: Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases. Word Scientific (2001)
2. Cpałka, K.: A New Method for Design and Reduction of Neuro-Fuzzy Classification Systems. IEEE Transactions on Neural Networks 20(4), 701–714 (2009)
3. Cpałka, K.: On evolutionary designing and learning of flexible neuro-fuzzy structures for nonlinear classification. Nonlinear Analysis Series A: Theory, Methods and Applications 71(12), e1659–e1672 (2009)
4. Cpałka, K., Rutkowski, L.: A new method for designing and reduction of neuro-fuzzy systems. In: 2006 IEEE International Conference on Fuzzy Systems, pp. 1851–1857 (2006)
5. Curkovic, M., Jezernik, K., Horvat, R.: FPGA-Based Predictive Sliding Mode Controller of a Three-Phase Inverter. IEEE Transactions on Industrial Electronics 60(2), 637–644 (2013)
6. Fogel, D.B.: Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, 3rd edn. IEEE Press, Piscataway (2006)
7. Gabryel, M., Cpałka, K., Rutkowski, L.: Evolutionary strategies for learning of neuro-fuzzy systems. In: I Workshop on Genetic Fuzzy Systems, Genewa, pp. 119–123 (2005)
8. Gabryel, M., Rutkowski, L.: Evolutionary Learning of Mamdani-Type Neuro-fuzzy Systems. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 354–359. Springer, Heidelberg (2006)
9. Gabryel, M., Rutkowski, L.: Evolutionary methods for designing neuro-fuzzy modular systems combined by bagging algorithm. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 398–404. Springer, Heidelberg (2008)

10. Hsu, C., Juang, C.: Continuous ant optimized type-2 fuzzy controller for accurate mobile robot wall-following control. In: International Conference on Fuzzy Theory and it's Applications (iFUZZY), 2012, pp. 187–191 (2012)
11. Li, X., Er, M.J., Lim, B.S., et al.: Fuzzy Regression Modeling for Tool Performance Prediction and Degradation Detection. International Journal of Neural Systems 20(5), 405–419 (2010)
12. Kaufman, Y., Ellenbogen, A., Meir, A., Kadmon, Y.: Nonlinear neural network controller for thermal treatment furnaces. In: IEEE 27th Convention of Electrical & Electronics Engineers in Israel (IEEEI), pp. 1–4 (2012)
13. Korytkowski, M., Gabryel, M., Rutkowski, L., Drozda, S.: Evolutionary methods to create interpretable modular system. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 405–413. Springer, Heidelberg (2008)
14. Koza John, R., Streeter Matthew, J., Keane Martin, A.: Routine high-return human-competitive automated problem-solving by means of genetic programming. Information Sciences 178(23), 4434–4452 (2008)
15. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer (1999)
16. Katsuhiko, O.: Modern Control Engineering. Prentice Hall (2001)
17. Przybył, A.: Doctoral dissertation: Adaptive observer of induction motor using artificial neural networks and evolutionary algorithms. Poznan University of Technology (2003) (in polish)
18. Przybył, A., Cpałka, K.: A new method to construct of interpretable models of dynamic systems. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS (LNAI), vol. 7268, pp. 697–705. Springer, Heidelberg (2012)
19. Przybył, A., Smolag, J., Kimla, P.: Real-time Ethernet based, distributed control system for the CNC machine. Electrical Review 2010-2 (2010) (in polish)
20. Rasoanarivo, I., Brechet, S., Battiston, A., Nahid-Mobarakeh, B.: Behavioral Analysis of a Boost Converter with High Performance Source Filter and a Fractional-Order PID Controller. In: IEEE Industry Applications Society Annual Meeting (IAS), pp. 1–6 (2012)
21. Rutkowski, L.: Computational Intelligence. Springer (2007)
22. Rutkowski, L., Cpałka, K.: Flexible weighted neuro-fuzzy systems. In: Proceedings of the 9th Neural Information Processing, pp. 1857–1861 (2002)
23. Rutkowski, L., Przybył, A., Cpałka, K., Joo, E.M.: Online Speed Profile Generation for Industrial Machine Tool Based on Neuro-Fuzzy Approach. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part I. LNCS, vol. 6113, pp. 645–650. Springer, Heidelberg (2010)
24. Rutkowski, L., Przybył, A., Cpałka, K.: Novel Online Speed Profile Generation for Industrial Machine Tool Based on Flexible Neuro-Fuzzy Approximation. IEEE Transactions on Industrial Electronics 59(2), 1238–1247 (2012)
25. Teja Ravi, A.V., Chakraborty, C., Maiti, S., Hori, Y.: A New Model Reference Adaptive Controller for Four Quadrant Vector Controlled Induction Motor Drives. IEEE Transactions on Industrial Electronics 59(10), 3757–3767 (2012)

The Niching Mechanism in the Evolutionary Method of Path Planning

Roman Śmierzchalski, Piotr Kolendo, Lukasz Kuczkowski, Bartosz Jaworski, and Anna Witkowska

Technical University of Gdańsk, Electrotechnics and Automation Department, Poland
pkolendo@ely.pg.gda.pl

Abstract. This paper presents the concept of the niching mechanism in the evolutionary method of path planning. The problem is considered based on the example of a ship path planning. In this method the diversity of individuals is tested in respect to their physical distance, not the fitness function value. The researches show that such an approach increases effectiveness of solution space exploration, what results in a final solution with a better fitness function value. This paper examines several sea collision scenarios at different levels of difficulty. Based on those, the method has been tested to choose values of parameters, which significantly influence its effectiveness.

Keywords: evolutionary algorithms, niching, path planning.

1 Introduction

Path planning of a moving object is a common theme in many technical applications such as: mobile robots [1,2] traffic regulated ship routing or a problem of avoiding collision at the sea [3,4,5]. The problem is defined as a task, where given a mobile object with certain dynamical and kinematical properties and an environment through which this object is travelling, one needs to plot a path between start and end points, which avoids all environments static and dynamic obstacles and meets the optimization criteria. In this article, the maritime environment with static (islands, constrained areas) and dynamic (other ships) constrains is taken as an example. Plotting the ships trajectory involves choosing a optimal route by the economic criterion that has to meet specific safety conditions. Considering the solutions in the previous works [3,4,5,6] the problem is being solved using adaptive evolutionary method [1,3]. The evolutionary path planning method is non-deterministic, based on a natural selection mechanism. Its most important advantages are build-on adaptation mechanism for a dynamic environment and reaching a multi-criteria task solution in a near-real time.

Based on Evolutionary Planner/Navigator (EP/N) [1], a modified version consisting of evolutionary algorithm library [7] (vEP/N++), has been developed. It takes into account the specific nature of a maritime collision scenarios.

This article proposes usage of niching mechanism for the problem of path planning based on the classical concept presented in [7,8]. The researches show

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 101–112, 2013.

that the usage of niching mechanism for environments populated with large number of dynamic obstacles leads to a final solution with a better value of the fitness function.

The article is organized as follows. In chapter 2 the evolutionary path planning algorithm (vEPN++) is presented. Chapter 3 presents the niching mechanism and the following chapter defines the simulation environment. Chapter 5 presents tests results and the final chapter concludes the paper.

2 Evolutionary Path Planning Method

According to transport plan an own ship should cover the given route in the determined time, on the other hand, it has to move safely along the planned trajectory while avoiding the navigational constrains and other moving objects. Path planning in a collision scenario has to stand a compromise between a deviation from a given course and ships safety. Thus the problem is defined as multi-criteria optimization task which considers safety and the economics of ships movement. Every path is evaluated based on the fitness function. In the considered case, the problem has been reduced to a single objective optimization task with weighting factors (1), (2), (3).

$$Total\_Cost(S) = Safe\_Cond(S) + Econ\_Cond(S) \tag{1}$$

$$Safe\_Cond(S) = w_c{}^* clear(S) \tag{2}$$

$$Econ\_Cond(S) = w_d{}^* dist(S) + w_s{}^* smooth(S) + w_t{}^* time(S) \tag{3}$$

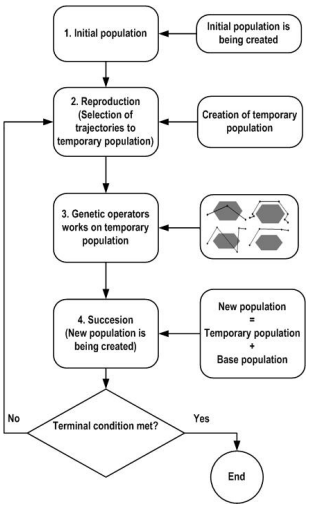


Fig. 1. Evolutionary algorithm cycle

Figure 1 shows a single cycle of evolutionary ship path planning algorithm. In the first step a random population is being initialized. In the second one, using a chosen selection scheme, a determined number of individuals is randomly selected to the temporary population. Then the genetic operators such as cross-over and mutation are working on the temporary population. In the next step a new population is established. It consists of best base and temporary population individuals. The algorithms cycle is repeated until the termination condition is met (a certain number of cycles in this instance). The presented algorithm is a Steady-State type.

Evolutionary algorithm searches an optimal solution, using a global exploration technique of the solution space and exploiting local extremes. The purpose of the exploration phase is to locate a set of disjoint areas of local extreme attraction. In the exploitation phase the areas of attraction for the local extremes are being searched. The balance between exploration and exploitation phases has a significant impact on the algorithm operation. Evolutionary algorithm should be able to leave an attraction area, on the other hand, it should be able to find the optimum quickly. Consequently, it is essential to establish a balance between the exploration and exploitation or, in other words, to keep the right value of the selective pressure. Selective pressure is a tendency of the algorithm to improve the mean value of the base population fitness function. The higher the selection pressure is, the number of expected copies of the better individuals is becoming greater in comparison to the number of the worse individuals.

3 Niching Mechanism

The niching mechanism in the problem of path planning is inspired by nature. It is based on the presumption that some species that are not well adapted in a specific moment of time, can develop independently. They will not become dominated if there is a certain geographical barrier (i.e. distance) separating the worse population from the actual better one. As a result the individuals that are not well adapted in actual moment of time, can be much better adapted later on, when dynamic changes in environent are take place. For the algorithmic sense, barrier means a distance between group of paths. Theoretically paths of lower fitness function value won't be dominated (lost) if the distance between them, according to the best individual, is large enough. In the later phase maintained paths can lead to achieve much better solutions, as algorithm preserve diversity of population members. As it was stated in chapter two, one of the main functions of the evolutionary algorithm is to control the population selective pressure, that is balancing the exploration and exploitation phase. As it was shown in article [10,11] maintaining the populations diversity has a positive impact on the quality of the solutions in the maritime environment, especially in a dynamically changing , congested areas. Achieving such diversity is possible by application of fitness function scaling, as it was presented in [10,11]. Fitness function scaling allows to control the selective pressure based on the fitness score of the individuals. In those cases diversity between individuals is being measured by modification

of fitness function value without taking into account physical distance between paths. This has an important impact on the search of the solution space, because the paths of a similar fitness function score can be placed far from one another and with the classical scaling techniques, the difference between them might not be visible. The proposed method based on the niching mechanism is also a method of fitness function scaling, however it also allows to compare physical distance between paths (based on the real distance, not the fitness function score). Such solution increases algorithms potential exploration abilities, because it allows to maintain an individual of slightly worse fitness function value, which is distant from the currently best exploited extremes. In case the niching mechanism is not used, such solution would be neglected. The niching mechanism used in this article is based on the classic concept from Goldbergs [8]. According to it, the modified fitness function f' (4), (5) has been introduced. The presented method allows to specify the diversity between individuals.

$$\mathbf{f}' = \frac{\mathbf{f}}{\sum\limits_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{s}(\mathbf{d_j})} \tag{4}$$

$$\mathbf{s}\left(\mathbf{d_j}\right) = \begin{cases} \mathbf{d_j} < \sigma \ \rightarrow \mathbf{1} - \left(\frac{\mathbf{d_j}}{\sigma}\right) \\ \mathbf{d_j} \geq \sigma \rightarrow \mathbf{0} \end{cases} \tag{5}$$

where:
f- fitness function,
n- members in population,
d- distance beetwen two members of population,
σ- size of the niche,
j- individual number.

The methods mechanism operates in the following way. To maintain the diversity of the individuals, niches are being formed by gathering "similar" solutions (by the distance, not fitness score criterion). Niches size are determined by the σ parameter, which is the maximum value at which an individual is considered as a member of a current collective. If the distance is greater than the value set by σ, a new niche is being created with similar principles. This is repeated for every member of the population. In a niche consisting of greater number of similar solutions, its fitness function value will be reduced because of the modification in equation (4). Such a solution allows to increase the probability of maintaining an individual of a theoretically worse fitness score, but distant from the other solutions proximate to the currently best (again certain criteria) one.

The effectiveness of the method is dependent on the choice of parameters presented in the equation (4). The most important issue is the way of calculating the distance between individuals in a niche. The considered task involves collision avoidance in the maritime environment, thus the key element in comparing two paths will be the spot, where the control object will be found in a specific moment of time t_i, which is shown on Figure 2. The comparison involves two individuals.

The calculation is being carried out as follows (6):

$$\mathbf{d}=\sum_{\mathbf{i=1}}^{\mathbf{n-1}}\mathbf{l_i} \tag{6}$$

Time needed to cover the path by own ship is being determined for both paths. In the next step the path with shorter value of time needed is selected and divided into equal time spans with the discretization step. On each of the path points corresponding to time spans t_i, lines l_i are plotted and then distance between them is calculated using Cartesian metric. The sum of the distances l_i from equation (6) is the total distance between the two individuals. This procedure is repeated for each population member and after that the fitness function value is calculated based on (4)).

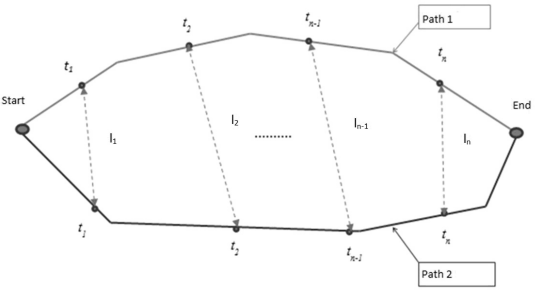


Fig. 2. Calculation of distance between two individuals

4 Simulation Environment

For the research purposes 5 example environments of different complexity (and different traffic load) have been used. For an evolutionary algorithm of fixed parameters, the impact of specific niching parameters (4) was being tested. In the next step the results were compared for several initial populations. The settings of the algorithm were as follows: crossing probability 0.7, mutation probability 0.2, population size 40, the amount of individuals exchanged in each generation 10. A Steady-State algorithm with partially exchangeable population was used for calculations. Based on the undertaken researches, the termination condition was set to 700 generations. This parameter is greater in comparison to earlier researches (400 generations in [10,11]). This is due to the fact, that usage of niching mechanism extends the duration of exploration phase, thus using the mechanism improves the solution space search process. Figures 3-7 present test collision avoidance scenarios.

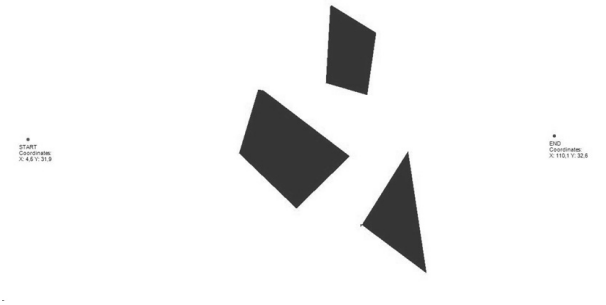


Fig. 3. 1-st simulation environment

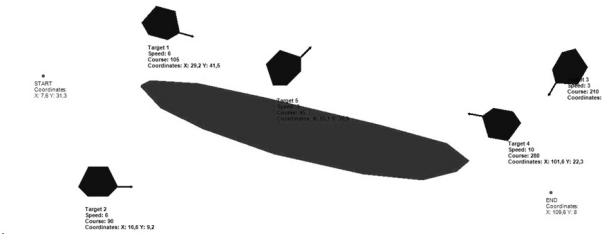


Fig. 4. 2-nd simulation environment

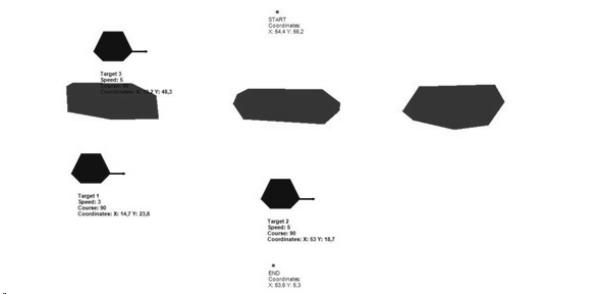


Fig. 5. 3rd simulation environment

5 Results

The partial results presented in tables 1 - 2 show general trend in the undertaken researches (introducing detailed results would cloud the trend visibility) by presenting the final fitness function value for simple environment (Table 1 for Figure 1)

and more complex one (Table 2 for Figure 6). Figures 8 - 9 show results of simulation for more complex cases where the advantages of nichinig mechanism are being outlined. Figures present simulation process for environments number 2 (Figure 4) and 5 (Figure 7).

Table 1. Partial results achieved in first environment

Environment 1							
Number of initial population		**1**	**2**	**3**	**4**	**5**	**6**
Size of the niche (σ)	40	218	234	303	328	259	303
	100	269	223	429	302	283	326
	1000	273	255	407	315	321	334
Number of points used in distance comparison (i)							
	3	257	210	222	218	266	265
	6	266	218	217	237	249	218
	7	266	245	288	218	358	276
	10	289	243	303	328	372	303
Number of generations, where niching mechanism is being used							
	0-700	276	295	340	218	220	222
	0-400	217	217	222	220	219	286
	0-200, 300-500	219	287	217	332	298	257
Results without niching mechanism		225	217	234	268	218	291

Results of researches will be discussed in correspondence to the most important aspects, which influence the effectiveness of the niching mechanism method.

– The size of the niche (σ) and the number of points used for calculating the distance between individuals (i).

In the first phase of researches the size of the niche and the number of points used for calculating the distance between individuals were determined. Those parameters are closely related to each other, because the number of points determines the total distance between individuals. Based on that distance, the algorithm checks if the path qualifies to the certain niche or if a new niche has to be created. According to the researches results, one can determine, that the

Table 2. Partial results achieved in 2-nd environment

Environment 1							
Number of initial population		**1**	**2**	**3**	**4**	**5**	**6**
Size of the niche (σ)	40	293	345	432	328	436	383
	100	369	415	421	344	389	473
	1000	449	517	513	522	490	530
Number of points used in distance comparison (i)							
	3	319	332	456	298	388	403
	6	347	329	530	368	418	390
	7	334	443	617	455	409	491
	10	544	708	480	603	457	508
Number of generations, where niching mechanism is being used							
	0-700	503	642	653	580	459	516
	0-400	289	314	415	352	320	387
	0-200, 300-500	670	340	459	613	318	433
Results without niching mechanism		493	690	789	953	519	845

number of points (defined by discrete time spans t_i presented in figure 2), used to compare distance between individuals, should be among 3 and 6. These values show the differences between paths in a most objective way, where exploration phase work as it was assumed in theory. Increasing number of points to higher values like 7-10 resulted in a noticeable decrease of both the solution space search capability and the final value as it is seen in Tables 1-2 and in figures 8-9. Furthermore, the size of the niche with a certain discrete step was being searched. The researches show that the niches with size of 15-20% of average path length have given the best performance (Table 1-2). By increasing the size of a niche the algorithms exploitation capabilities were decreased, because the solutions around them obtain much lower fitness function values. With the large niche size the algorithm losses both exploration and exploitation capabilities and the most of the paths land in the same niche and their competitiveness is averaged. Thus, the algorithm does not add new paths and does not exploit the existing extremes.

– The scope of the niching mechanism application.

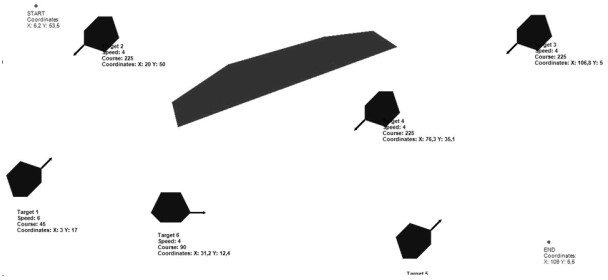


Fig. 6. 4-th simulation environment

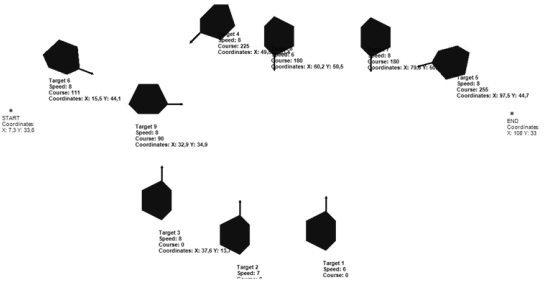


Fig. 7. 5-th simulation environment

The scope of application refers to the number of generations during which the niching mechanism should operate. The article presents three most important options (according to researches results): Usage of the mechanism throughout the whole generation process, usage with some initial generations and alternate usage (it is used for some generations, then it is disabled for few generations, to be reactivated later and so on). The results show that the usage of the mechanism throughout the whole generations has an adverse affect on the final solution values. As it was already mentioned, due to the application of niching mechanism, the exploitation process is weakened, thus despite having a favorable search capabilities of solution space, the algorithm will have problems with exploitation of found extremes. The results from the alternate method were unstable (some final results has improved, some get worsen, without a stable trend), that is why it is not recommended. The researches show (Tables 1-2) that the best results are achieved when the mechanism is operating for the first 70% of the generations. For the remaining 30%, the algorithm should focus on the exploitation of the existing niches, therefore the mechanism should be turned off.

The greatest benefit offered by the niching mechanism is the increase of algorithm exploration capabilities. Without it the algorithm usually converges around single local extreme and exploits it. When using the niching variant, an alternative solution is almost always present (Figures 8- 9). As it is seen on those

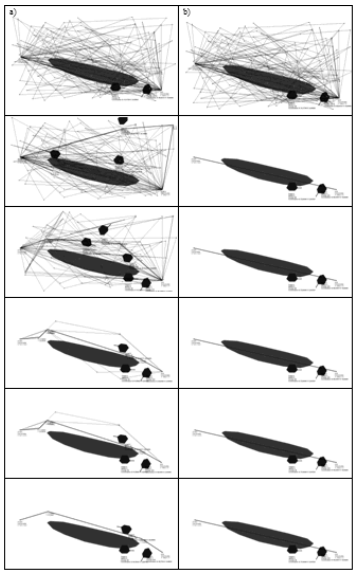


Fig. 8. a) Environment 2, with application of niching mechanism, population after 0, 50, 150, 250 ,400 ,700 generations, b) Environment 2, without application of niching mechanism, population after 0, 50, 150, 250 ,400 ,700 generations.

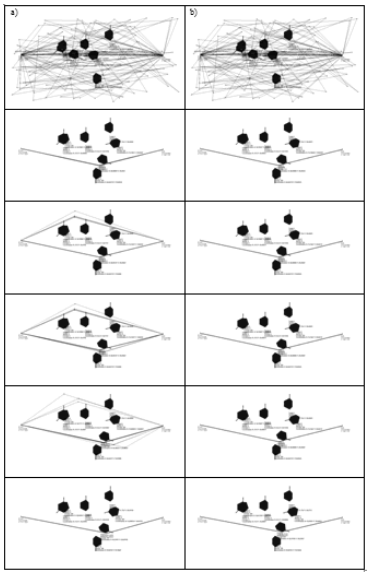


Fig. 9. a) Environment 5, with application of niching mechanism, population after 0, 50, 150, 250 ,400 ,700 generations, b) Environment 5, without application of niching mechanism population after 0, 50, 150, 250 ,400 ,700 generations.

figures the diversity is maintained much longer in comparison to the case where the niching mechanism was not used. In the case of less complex environments [1] usage of the niching mechanism does not noticeably impact the final average fitness score (nor improve, nor worsen). This is due to the fact the simple environments offer less alternative solutions and there is usually a single dominant niche, where the exploitation process takes place.

The situation is quite different when the environment is more complex (environments 2, 3, 4, 5), especially with large number of dynamic constrains. Algorithm without niching mechanism achieves significantly worse values of final fitness function and in some cases is not able to find a collision free route (Figures 8 - 9).It is clearly seen on figures 8a, 8b, where the diversity of the population after 250 generations is shown. The niche-less algorithm falls into the local minimum too quickly and has problems with leaving it. The example environment 2,5 shows that because of the niching mechanism, it is possible to find a better solution due to improved exploitation capabilities of the algorithm. The comparison of the algorithm's calculations (intermediate and final results) are presented on figures 8 - 9.

6 Conclusions

The application of the niching mechanism allows to increase the exploration abilities of the algorithm by comparing the mutual distance between path in the Cartesian coordinate system. Less complex environment [1] does not benefit from the niching mechanism as it achieves similar final fitness function values as without it. In case of more complex environments, algorithm without application of niching mechanism performs worse and in some cases it was not able to find acceptable solution, as it was shown in chapter 5. For complicated navigational situations a problem might appear, in case when only unfeasible solutions are present. Until the first feasible path is found, the algorithm tends to generate complicated unfeasible solutions and is not being able to explore the solution space. The best response for that problem is to increase the number of generations. In all cases it helps the algorithm to find feasible solution and then work properly (as it was assumed in theory). For optimal use of the mechanism it is necessary to increase the number of algorithm generations in comparison to a situation where niching is not being used. This will increase the calculation time, however the results will still be obtained in near real time. The calculations will be longer around 15-20% . The results presented in chapter 5 allow to conclude, that usage of niching mechanism in the evolutionary path planning increases exploration capabilities of the algorithm, what result in finding final solution with better fitness function value.

The proposed method of trajectory comparison (niching mechanism) can be used not only in problem of collision avoidance at sea, but also in other cases (i.e. the path planning for mobile robots). The algorithms recommended settings do not correspond to the dynamic of a specific object, so it is possible to use it in different applications.

Acknowledgement. The authors would like to thank the Polish Ministry of Science and Higher Education for funding this research under grant no. N514 472039.

References

1. Xiao, J., Michalewicz, Z., Zhang, L., Trojanowski, K.: Adaptive Evolutionary Planner/Navigator for Mobile Robots. IEEE Transactions on Evolutionary Computation 1(1), 18–28 (1997)
2. Yap, C.K.: Algorithmic Motion Planning. In: Schwartz, J.T., Yap, C.K. (eds.) Advances in Robotics. Algorithmic and Geometric Aspects of Robotics, vol. 1, pp. 95–143. Lawrence Erlbaum Associates (1987)
3. Smierzchalski, R.: Trajectory planning for ship in collision situations at sea by evolutionary computation. In: Proceedings of the IFAC MCMC, Brijuni, Croatia (1997)
4. Smierzchalski, R., Michalewicz, Z.: Modeling of a Ship Trajectory in Collision Situations at Sea by Evolutionary Algorithm. IEEE Transaction on Evolutionary Computation 4(3), 227–241 (2000)
5. Smierzchalski, R., Michalewicz, Z.: Path Planning in Dynamic Environments. In: Patnaik, S., Jain, L.C., Tzafestas, S.G., Resconi, G., Konar, A. (eds.) Innovations in Machine Intelligence and Robot Perception. SCI, vol. 8, pp. 135–154. Springer, Heidelberg (2005)
6. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer (1996)
7. Wall, M.: GAlib: A C Library of Genetic Algorithm Components. MIT (1996)
8. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Longman Publishing Co. Inc., Boston (1989)
9. Arabas, J.: Lectures on Evolutionary Algorithms. Technical and Scientific Publishing, Warsaw, Poland (2004) ISBN 978-8-320-42970-1
10. Kolendo, P., Smierzchalski, R., Jaworski, B.: Experimental research on evolutionary path planning algorithm with fitness function scaling for collision scenarios. In: Methods and Algorithms in Navigation Marine Navigation and Safety of Sea Transportation, Gdynia, Poland, pp. 85–91 (2011) ISBN 978-0-415-69114-7
11. Kolendo, P., Smierzchalski, R., Jaworski, B.: Scaling Fitness Function in Evolutionary Path Planning Method. In: 20th IEEE International Symposium on Industrial Electronics Proceedings. IEEE-ISIE (2011)

New Algorithm for Evolutionary Selection of the Dynamic Signature Global Features

Marcin Zalasiński, Krystian Łapa, and Krzysztof Cpałka

Częstochowa University of Technology,
Institute of Computational Intelligence, Poland
{marcin.zalasinski,krystian.lapa,krzysztof.cpalka}@iisi.pcz.pl

Abstract. Methods using dynamic signature for identity verification may be divided into three main categories: global methods, local function based methods and regional function based methods. Global methods base on a set of global parametric features, which are extracted from signature of user. Global feature extraction methods have been often presented in the literature. Another interesting task is selection of a features group which will be considered individually for each user during training and verification process. In this paper we propose a new approach to automatic evolutionary selection of the dynamic signature global features. Our method was tested with use of the SVC2004 public on-line signature database.

1 Introduction

Biometrics is the science of recognizing the identity of a person based on some kind of unique personal attributes. The attributes may be physical like a fingerprint, a face (see e.g. [14], [25]-[26]) or behavioural like a gait, a signature (see e.g. [8], [30]-[31]) etc. Dynamic signature is a kind of signature biometric attribute, which contains not only information about trajectory, but also information about the dynamics of the signature. Signature may be treated as a unique identifier of the user, used during identity verification process (see e.g. [10], [30]-[31]).

The dynamic signature verification methods may be categorized into three main groups (see e.g. [22]): global methods, local function based methods and regional function based methods. In global methods a set of features are extracted from the signature (e.g. signature total duration, number of pen-downs) and classification process is performed on the basis of these features. Local function based methods use the time functions of the signatures, which are matched and compared. One of the methods used for this purpose is Dynamic Time Warping (see e.g. [28]). In regional function based approaches some regional properties are estimated from the signatures and then they are used to train classifier.

In this paper we focus on the global approach. In the literature a few sets of features have been proposed (see [18], [23], [24]). Some features have been taken from the literature relating to identity verification based on signature but other have been created by authors on the basis of their experience in the signature

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 113–121, 2013.

acquisition procedure. In [9] the authors have collected all features from [18], [23], [24], some of them have been adapted and some new features have been add to this set. In this paper we base on the feature set proposed in [9]. It should be noted that the operation of our method is not dependent on the adopted feature set. The feature set can be practically arbitrarily reduced or extended.

Large global feature set may be reduced by selection of optimal features subset, which will be considered during classification phase. In this case global features may be ranked and only features with the highest rank value are used in classification process. Approaches to the rate of global features have been presented in [9] and [18]. In [18] authors propose approach based on computation of rank using mean value and standard deviation of the feature. Rank for each feature is calculated separately. In [9] the method based on the distance between the mean value of the feature from training signatures of the user and the feature value of all training signatures from all users is presented. This method performs the overall ranking of features, without detailing the importance of features for the user.

In this paper we propose a new method for selection of the dynamic signature global features. The method is based on a modified genetic algorithm which is used to select of features set used in the verification process. Idea of genetic algorithms has been taken from the natural process in which live organisms transfer their features to their offspring during creation of new population. The algorithms may be applied to solve various optimization problems. In the method presented in this paper features are selected individually for each signer. Our algorithm is characterized by a special way of determining the fitness function. It bases on classification with use of the Principal Component Analysis method (see e.g. [19]) and so called reconstruction error.

This paper is organized into four sections. In Section 2 we present novel algorithm for evolutionary selection of the dynamic signature global features. In Section 3 simulation results are presented. Conclusions are drawn in Section 4.

2 Idea of the New Method of Features Selection

Method proposed in this paper uses combination of genetic algorithm (see e.g. [2]-[7], [11], [16], [17]) and PCA method for selection of global features subset used during verification process. The selection process is performed individually for each user, so each subset of features is unique for the signer. Subset of features is chosen from all one hundred features presented in [9]. Examples of features which belong to the set of features in [9] are shown in Table 1.

Block diagram of the proposed algorithm is presented in Fig. 1 a). The algorithm selects the subset of global features and creates classifier for the i-th user. First, selection of training signatures for i-th user is performed (**block 1**). The next random selection of the other users signatures is performed (**block 2**). This signatures are treated as forgeries. This approach commonly appears in the literature in the field of identity verification based on the dynamic signature (see e.g. [29]). In the third step, selection of the optimal global features

Table 1. Examples of features used in [9]

Feature description	Symbol
signature total duration	T_s
number of pen-ups	N(pen-ups)
standard deviation of y-trajectory acceleration	σ_{a_y}
standard deviation of y-trajectory velocity	σ_{v_y}
mean value of velocity divided by maximum value of velocity	$\bar{v}/v_{\max}$

subset for the i-th user is performed (**block 3**). For this purpose the algorithm *GeneticFeaturesSelection()* described in Section 2.1 and presented in Fig. 1 b) is used. Next, the algorithm returns the subset of genuine training signatures which are used during creation of the classifier (**block 4**) and subset of global features selected for the i-th user (**block 5**). The data from block 4 and block 5 are used during the verification phase.

2.1 Genetic Features Selection

Schema of genetic features selection is presented in Fig. 1 b). The algorithm selects the chromosome encoding the best subset of global features for the i-th user. In the first step the creation of an initial population is performed, which consists of the random selection of chromosomes (**block 1**). Each chromosome contains information about selected subset of features, each gene in the chromosome represents one global feature. Length of the chromosome corresponds to the number of all features. Each chromosome is represented by a binary sequence. If the value of the gene is "1", the feature which number is equal to the number of a gene's position is selected for the subset of the features represented by the chromosome. Otherwise if the value of gene is "0", the feature which number is equal to the number of a gene's position is not selected for the subset of the features represented by the chromosome. Next, the evaluation of the fitness of chromosomes in a population is performed (**block 2**). During this step value of the fitness function for each chromosome is calculated. It is realized in a separate algorithm, called *CalculateFf()*, presented in Fig. 2 a). In our algorithm the lower value of this function means the better the "quality" of the chromosome. In determination of fitness function value PCA method is used for computation of reconstruction errors. The errors decide about quality of subset of features encoded in the chromosomes for the i-th user. The detailed description of the fitness function used in our algorithm is presented in Section 2.2. Next, the stopping criterion is checked (**block 3**). If the stopping criterion is not met then the next step is selection of chromosomes (**block 4**). The selection of chromosomes consists in selecting, based on the calculated values of the fitness function. Selected chromosomes will take part in the creation of offspring until the next generation. The chromosomes having the best value of the fitness function have the most of the chances for the participation in the creation of new individuals. The selection can be performed using one of the methods

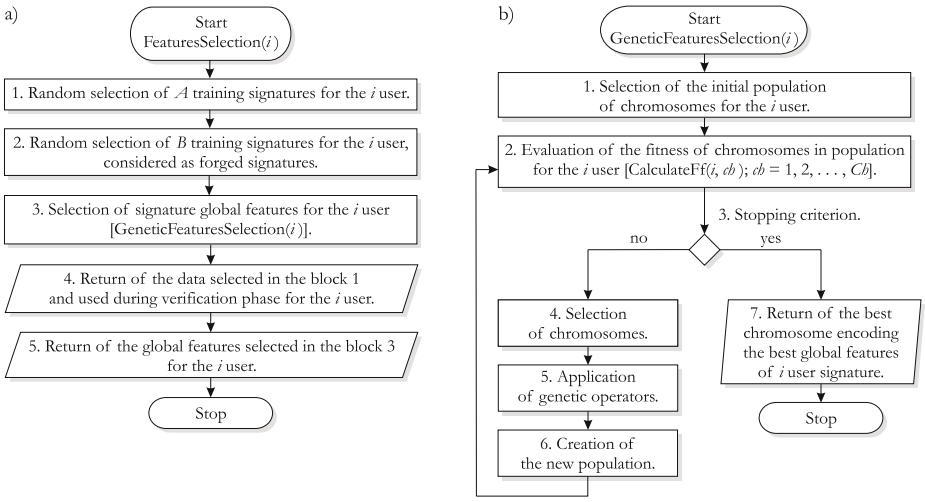


Fig. 1. Block diagram of: a) new method of features selection, b) genetic features selection

available in the literature (see e.g. [27]). Next, application of genetic operators is performed (**block 5**). The operators are applied to the chromosomes selected in 4-th step. It leads to the creation of a new population constituting the offspring population derived from the parents population. In our algorithm standard operators for genetic algorithms are used - crossover and mutation (see e.g. [1], [20]). In the next step creation of a new population is performed (block 6). The chromosomes obtained as a result of the operation of genetic operators belong to a new population and this population becomes the so-called current population for a given generation of the genetic algorithm. Next, stopping criterion is checked (**block 3**). If the stopping condition is met, "best" chromosome, which has the best fitness function value, is presented (**block 7**). In our algorithm the "best" chromosome contains information about subset of global features of the i user which will be considered during verification phase.

2.2 Fitness Function Determination

Schema of the fitness function used in our algorithm is presented in Fig. 2 a). The algorithm calculates fitness of the chromosome. First, selection of C signatures from A training signatures of the i-th user is performed (**block 1**). Next, PCA classifier is created (**block 2**). The classifier is created on the basis of the signatures from the previous step and features encoded in the chromosome for which fitness is calculated. In the next step (**block 3**) reconstruction error for the remaining genuine training signatures $errg_{i,j,ch}$, $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, A - C$, $ch = 1, 2, \ldots, Ch$, is computed from the following equation:

$$errg_{i,j,ch} = \left\| \mathbf{f}_{i,j,ch} - \left(\mathbf{E}_{i,ch} \mathbf{E}_{i,ch}^{T} \right) \mathbf{f}_{i,j,ch} \right\|, \tag{1}$$

where $(\mathbf{f}_{i,j,ch})_{p\times 1}$, $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, A-C$, $ch = 1, 2, \ldots, Ch$, contains features encoded in ch-th chromosome for j-th signature of i-th user, $\mathbf{E}_{i,ch}$, $i = 1, 2, \ldots, I$, $ch = 1, 2, \ldots, Ch$, contains the eigenvectors of the covariance matrix created on the basis of the features from C training signatures selected in the block 1. In this step reconstruction error for the forged training signatures $errf_{i,j,ch}$, $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, B$, $ch = 1, 2, \ldots, Ch$ is also computed analogously. This operation is performed with use of the PCA classifier created in the block 2. Next, selection of maximum value of reconstruction error for genuine training signatures, $Rg_{\max}$, is performed (**block 4a**) and selection of minimum value of reconstruction error for forged training signatures, $Rf_{\min}$, is performed (**block 4b**). In the next step fitness function of the chromosome is determined (**block 5**). Value of this function is calculated from equation:

$$\mathrm{ff}(ch) = \frac{Rgmax_{i,ch}}{Rfmin_{i,ch}}, \tag{2}$$

where ch is chromosome for which fitness function is determined, $Rgmax_{i,ch}$, $i = 1, 2, \ldots, I$, $ch = 1, 2, \ldots, Ch$, is maximum value of reconstruction error for genuine training signatures of i-th user determined by the following equation:

$$Rgmax_{i,ch} = \max\{errg_{i,1,ch}, errg_{i,2,ch}, \ldots, errg_{i,M-O,ch}\} \tag{3}$$

and $Rfmin_{i,ch}$, $i = 1, 2, \ldots, I$, $ch = 1, 2, \ldots, Ch$, is minimum value of reconstruction error for forged training signatures of i-th user determined by the following equation:

$$Rfmin_{i,ch} = \min\{errf_{i,1,ch}, errf_{i,2,ch}, \ldots, errf_{i,N,ch}\}. \tag{4}$$

In the last step of this algorithm value of the fitness function for the chromosome is returned (**block 6**).

2.3 Identity Verification Phase

The process of identity verification based on the dynamic signature is presented in Fig. 2 b). First, test signature is acquired by the system (**block 1**). Next, PCA classifier is created (**block 2**). The classifier is created on the basis of features encoded in the best chromosome and C training signatures of the user which identity is verified. Next, verification is performed with use of the one-class PCA classifier (**block 3**). In the last step of the algorithm, the result of identity verification is presented (**block 4**). Identity is verified if the following equation is satisfied:

$$errtest_i \geq cth_i, \tag{5}$$

where $errtest_{i,ch}$ is reconstruction error of test signature computed on the basis of features encoded in the best chromosome from block 7 of *GenuineFeatures-Selection()* function, i is a number of user which identity is verified and cth_i is a parameter determined individually for each user during training phase. Value

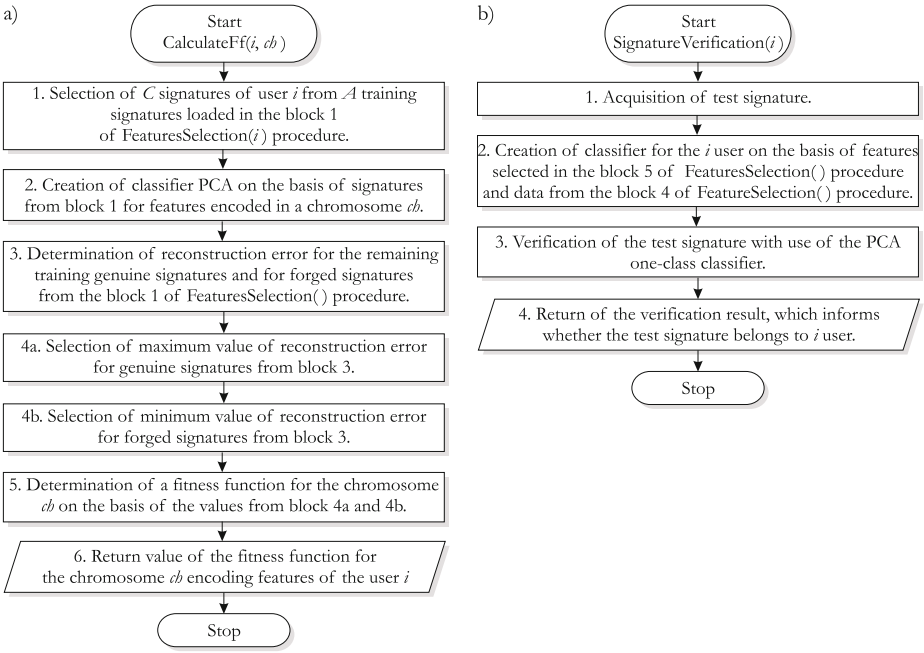


Fig. 2. Block diagram of: a) fitness function, b) signature verification

for this parameter is chosen to minimize the value of FAR and FRR errors and the difference between them.

3 Simulation Results

Simulations were performed using SVC 2004 public database (see [29]). During the simulation the following assumptions have been adopted:

- population contains 100 chromosomes,
- algorithm stops after the lapse of a determined number of 1000 generations,
- during selection of chromosomes tournament selection method is used,
- crossover is performed with probability equal to 0.8 at three points,
- mutation is performed for each gene with probability equal to 0.02
- number of all genuine training signatures $A = 5$, number of forged training signatures $B = 5$, number of genuine training signatures used for training PCA classifier in the *CalculateFf()* function $C = 2$.

The database contains 40 signers and for each signer 20 genuine and 20 forgery signatures. The test was performed five times, every time for all signers stored in the database. During test phase 10 genuine signatures (numbers 11-20) and 20 forgery signatures (numbers 21-40) of each signer were used. Simulations were performed in the authorial environment implemented in C#.

During simulation we tested two methods of verification based on global features. The first one was our method described in this paper. The second was the random subspace method proposed in [12] and used in [21]. The method generates B new training sets of features, builds classifiers on the basis of these sets, and finally combines them into a final decision rule. In our simulations PCA one class classifiers were used.

Results of the simulations are presented in the Table 1. The table contains values of FAR (False Acceptance Rate) and FRR (False Rejection Rate) errors which are commonly used in the literature to evaluate the effectiveness of identity verification methods (see e.g. [13]-[15]).

Table 2. Results of simulation performed by our system

Method	**Average FAR**	**Average FRR**	**Average error**
Random subspace method [12]	25.75 %	24.60 %	25.18 %
Our method	**23.87 %**	**22.65 %**	**23.26 %**

Conclusions of the simulations can be summarized as follows:

- The accuracy of our method is higher in comparison to the method described in [12].
- Our method, in contrast to the method given in [12], in the process of classification uses a single set of attributes selected during training phase.
- The advantage of the proposed method is that it allows to characterize the individual user.
- It also allows to extract the features which are often selected as reliable in the context of all users.

4 Conclusions

In this paper a new method for evolutionary selection of the dynamic signature global features is presented. The method assumes selection of the subset of global features from a large set of the features. This process is performed using genetic algorithm with PCA. The features are selected individually for each user. The achieved accuracy of signature verification in comparison with the other method proves correctness of the assumptions. In the future we will seek to determine weights of importance for each feature in context of the user which will be also considered during the verification process.

Acknowledgment. The project was financed by the National Science Center on the basis of the decision number DEC-2011/01/N/ST6/06964.

References

1. Arabas, J.: Lectures on Evolutionary Algorithms. Scientific-Technical Publishing House WNT (2001) (in polish)
2. Cpałka, K., Rutkowski, L.: A new method for designing and reduction of neuro-fuzzy systems. In: IEEE Int. Conference on Fuzzy Systems, pp. 1851–1857 (2006)
3. Cpałka, K.: A method for designing flexible neuro-fuzzy systems. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 212–219. Springer, Heidelberg (2006)
4. Cpałka, K., Rutkowski, L.: Neuro-fuzzy structures for pattern classification. WSEAS Trans. on Computers 697–688 (2005)
5. Cpałka, K., Rutkowski, L.: Neuro-fuzzy systems derived from quasi-triangular norms. In: IEEE International Conference on Fuzzy Systems, vol. 2, pp. 1031–1036 (2004)
6. Cpałka, K.: On evolutionary designing and learning of flexible neuro-fuzzy structures for nonlinear classification. Nonlinear Analysis series A: Theory, Methods & Applications 71 (2009)
7. Dziwiński, P., Rutkowska, D.: Ant focused crawling algorithm. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 1018–1028. Springer, Heidelberg (2008)
8. Ekinci, M., Aykut, M.: Human Gait Recognition Based on Kernel PCA Using Projections. Journal of Computer Science and Technology 22, 867–876 (2007)
9. Fiérrez-Aguilar, J., Nanni, L., Lopez-Peñalba, J., Ortega-Garcia, J., Maltoni, D.: An On-Line Signature Verification System Based on Fusion of Local and Global Information. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 523–532. Springer, Heidelberg (2005)
10. Fierrez, J., Ortega-Garcia, J.: On-line signature verification. In: Jain, A.K., Ross, A., Flynn, P. (eds.) Handbook of Biometrics (2008)
11. Gabryel, M., Rutkowski, L.: Evolutionary methods for designing neuro-fuzzy modular systems combined by bagging algorithm. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 398–404. Springer, Heidelberg (2008)
12. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. 20(8), 832–844 (1998)
13. Jain, A.K., Griess, F.D., Connell, S.D.: On-line signature verification. Pattern Recognition 35, 2963–2972 (2002)
14. Jie, Y., Fang, Y.Y., Renjie, Z., Qifa, S.: Fingerprint minutiae matching algorithm for real time system. Pattern Recognition 39, 143–146 (2006)
15. Kholmatov, A., Yanikoglu, B.: Identity authentication using improved online signature verification method. Pattern Recognition Letters 26, 2400–2408 (2005)
16. Korytkowski, M., Gabryel, M., Rutkowski, L., Drozda, S.: Evolutionary methods to create interpretable modular system. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 405–413. Springer, Heidelberg (2008)
17. Li, X., Er, M.J., Lim, B.S., et al.: Fuzzy Regression Modeling for Tool Performance Prediction and Degradation Detection. International Journal of Neural Systems 20(5), 405–419 (2010)
18. Lee, L.L., Berger, T., Aviczer, E.: Reliable on-line human signature verification systems. IEEE Trans. on Pattern Anal. and Machine Intell. 18, 643–647 (1996)

19. Mazhelis, O.: One-class classifiers: a review and analysis of suitability in the context of mobile-masquerader detection. South African Computer Journal 36, 29–48 (2006)
20. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer (1998)
21. Nanni, L.: Experimental comparison of one-class classifiers for online signature verification. Neurocomputing 69, 869–873 (2006)
22. Nanni, L., Maiorana, E., Lumini, A., Campisi, P.: Combining local, regional and global matchers for a template protected on-line signature verification system. Expert Systems with Applications (2009)
23. Nelson, W., Kishon, E.: Use of dynamic features for signature verification. In: Proc. of the IEEE Intl. Conf. on Systems, Man, and Cyber., vol. 1, pp. 201–205 (1991)
24. Nelson, W., Turin, W., Hastie, T.: Statistical methods for on-line signature verification. Intl. Journal of Pattern Recognition and Artificial Intell. 8, 749–770 (1994)
25. Pabiasz, S., Starczewski, J.T.: Face reconstruction for 3D systems. In: Selected Topics in Computer Science Applications, pp. 54–63. EXIT (2011)
26. Pabiasz, S., Starczewski, J.T.: Meshes vs. depth maps in face recognition systems. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part I. LNCS (LNAI), vol. 7267, pp. 567–573. Springer, Heidelberg (2012)
27. Rutkowski, L.: Computational intelligence. Springer (2007)
28. Tormene, P., Giorgino, T., Quaglini, S., Stefanelli, M.: Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. Artificial Intelligence in Medicine 45, 11–34 (2009)
29. Yeung, D.-Y., Chang, H., Xiong, Y., George, S., Kashi, R., Matsumoto, T., Rigoll, G.: SVC2004: First International Signature Verification Competition. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, pp. 16–22. Springer, Heidelberg (2004)
30. Zalasiński, M., Cpałka, K.: A new method of on-line signature verification using a flexible fuzzy one-class classifier. In: Selected Topics in Computer Science Applications, pp. 38–53. EXIT (2011)
31. Zalasiński, M., Cpałka, K.: Novel algorithm for the on-line signature verification. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS (LNAI), vol. 7268, pp. 362–367. Springer, Heidelberg (2012)

On the Performance of Master-Slave Parallelization Methods for Multi-Objective Evolutionary Algorithms*

Alexandru-Ciprian Zăvoianu[1,3], Edwin Lughofer[1], Werner Koppelstätter[3,4], Günther Weidenholzer[3,4], Wolfgang Amrhein[2,3], and Erich Peter Klement[1,3]

[1] Department of Knowledge-based Mathematical Systems/Fuzzy Logic Laboratory Linz-Hagenberg, Johannes Kepler University of Linz, Austria
[2] Institute for Electrical Drives and Power Electronics, Johannes Kepler University of Linz, Austria
[3] ACCM, Austrian Center of Competence in Mechatronics, Linz, Austria
[4] LCM, Linz Center of Mechatronics, Linz, Austria

Abstract. This paper is focused on a comparative analysis of the performance of two master-slave parallelization methods, the basic *generational* scheme and the *steady-state asynchronous* scheme. Both can be used to improve the convergence speed of multi-objective evolutionary algorithms (MOEAs) that rely on time-intensive fitness evaluation functions. The importance of this work stems from the fact that a correct choice for one or the other parallelization method can lead to considerable speed improvements with regards to the overall duration of the optimization. Our main aim is to provide practitioners of MOEAs with a simple but effective method of deciding which master-slave parallelization option is better when dealing with a time-constrained optimization process.

Keywords: evolutionary computation, multi-objective optimization, performance comparison, master-slave parallelization, steady-state evolution.

1 Introduction and State-of-the-Art

Many real-world optimization problems are multi-criteria by nature and usually involve several conflicting objectives (e.g. cost vs. quality, risk vs. return on investment). Problems falling within this class are referred to as *multi-objective optimization problems* (MOOPs in short) and, generally, such problems do not have a single solution. Solving them requires finding a set of non-dominated solutions called the *Pareto-optimal set*. Each solution in this set is better than any other solution in the set with regards to at least one optimization objective (i.e., it is not fully dominated by another solution). The objective space representation of the Pareto-optimal set is called the *Pareto front*.

Multi-objective evolutionary algorithms (MOEAs) have proven to be one of the most successful soft computing techniques for solving MOOPs [2]. This is because of their

* This work was conducted in the realm of the research program at the Austrian Center of Competence in Mechatronics (ACCM), which is a part of the COMET K2 program of the Austrian government. The work-related projects are kindly supported by the Austrian government, the Upper Austrian government and the Johannes Kepler University Linz. The authors thank all involved partners for their support. This publication reflects only the authors' views.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 122–134, 2013.

inherent ability to produce complete Pareto-optimal sets over single runs. As most stochastic methods, MOEAs are approximate methods that cannot guarantee finding the strictly optimal solution set (i.e., the *true Pareto front* of the problem), but they are fairly flexible and robust and can find high quality non-dominated solution sets in a reasonable amount of time.

The main drawback of MOEAs is the fact that they require a large number of solutions to be evaluated during an optimization run. The issue can become particularly problematic for optimization problems that require very time intensive fitness evaluation functions in order to compute objective or constraint values. In such cases, optimization runs can last for several days, see for instance the approach in [13] where MOEAs are used for the optimization of combustion in a diesel engine and the approach in [17] applied for optimizing design parameters of electrical drives.

This problem of very lengthy optimization runs is usually alleviated by the parallelization and distribution of the MOEA over a computer cluster or grid environment. There are several paradigms (architectural and/or conceptual models) of parallelizing a MOEA: master-slave, island, diffusion, hierarchical and hybrid models (please see Chapter 8 of [1]). The most straight forward and easiest to implement parallelization method for evolutionary algorithms is the master-slave model: fitness evaluations are distributed between several slave nodes, while all the evolutionary operations (selection, crossover, mutation) are performed on a master node. The master-slave parallelization is suitable both for a classic, generational approach, as well as for an asynchronous parallelization approach similar to the steady-state selection scheme [10].

In several real-world master-slave parallelization setups for MOEAs, the duration of the sequential computation tasks is also very important. This usually happens because of some rather lengthy pre-evaluation steps that must be performed locally on the master node for each generated individual, before dispatching the individual for remote fitness evaluation on the slave nodes. The reasons for performing these pre-evaluation steps on the master node may include security concerns, software licensing issues, network configuration settings, etc. It is fairly intuitive that whenever the average duration of the sequential task carried out on the master node is significant with regards to the average duration of the fitness evaluation task, the speed-up that can be achieved by using a parallel / distributed hardware architecture is affected (q.v. Amdahl's law).

A recent study indicates that, for optimization problems with a heterogeneous (non-constant) time-wise fitness distribution, the steady state asynchronous parallelization is somewhat better in terms of convergence (Pareto quality and global run-time) than the generational approach [13]. In [8], Durillo et al. also show evidence that applying a steady state approach can bring improvements in terms of Pareto quality. The present research builds on these earlier findings and tries to determine the reasons that might influence the average performance of the two master-slave parallelization schemes in the context of MOEAs. Our main intention is to help practitioners in this field to decide what is the most efficient parallelization option based on the particularities of their concrete optimization scenarios. The comparison is especially focused on the practical aspect of Pareto quality / run-time performance: *which parallelization method is more likely to deliver the highest quality solution in a pre-defined global run-time interval*.

The tests we report on were performed taking into consideration two of the most widely used MOEAs: the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [4] and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [15]. At a high level of abstraction, NSGA-II and SPEA2 are in fact different MOOP orientated implementations of the same concept - the $(\mu + \lambda)$ evolutionary strategy (where μ denotes the archive size and λ the population size). The two main features of both algorithms are *a*) a highly elitist approach that is based on an archive population that stores the best individuals found during the run; *b*) a two-tier fitness evaluation function that uses a primary Pareto non-dominance metric and a secondary solution density estimation metric.

2 The Considered Master-Slave Parallelization Schemes

The diagram in Figure 1 provides a general explanation of the computation cycles used in both master-slave parallelization schemes. The basic principles of the two paralleliza-

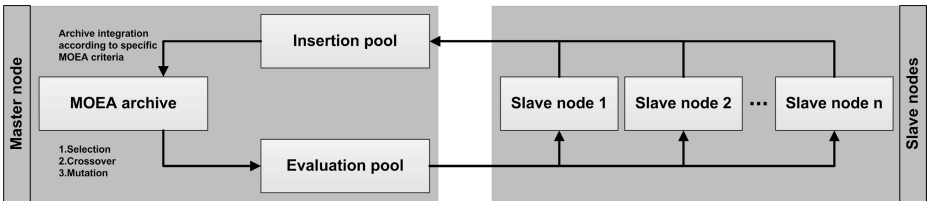


Fig. 1. Diagram of the GEN-MSPS and SSA-MSPS computation cycle

tion schemes for a generic $(\mu + \lambda)$-archiving based MOEA are:

1. **The Generational Master-Slave Parallelization Scheme (GEN-MSPS)** - In this case, the computation cycle is regulated by a **synchronization step**. This step occurs at the integration of individuals from the *Insertion pool* into the archive. The master node must block until all the λ individuals of the current population have been evaluated on the slave nodes. After this requirement is satisfied, the specific $(\mu + \lambda)$-archiving algorithm is used in order to update the *MOEA archive* on the master node. Afterwards, all the λ individuals of the next generation are created sequentially on the master node using the specific genetic operators (selection, recombination and mutation) and inserted into the *Evaluation pool*. Each slave node asynchronously selects an individual from this pool, evaluates it and afterwards places the results in the *Insertion pool*. The procedure described above repeats itself until the optimization stopping criterion is met. From an algorithmic point of view, this parallelization scheme is identical to a sequential implementation.
2. **The Steady-State Asynchronous Master-Slave Parallelization Scheme (SSA-MSPS)** - In this case, the computation cycle is regulated only by the interplay between the computation time requirements of the different parts of the algorithm (i.e. fitness evaluation, generation of new individuals, archive update, etc.). The slave nodes operate in the same way as in the generational parallelization scheme.

The master node operates based on a very simple loop. While the stopping criterion is not met, the master node first checks if there is an evaluated individual in the insertion pool and if such an individual exists, it collects it and updates the *MOEA archive*. This is the main difference to GEN-MSPS, which in each cycle has to wait for the evaluation of *all* individuals before new individuals can be generated. Secondly, the master node creates one new individual and inserts it immediately into the *Evaluation pool*. The above computation cycle resembles classical steady-state selection as, at a given time, one new individual is created and one evaluated individual is collected and, if fit enough, inserted into the archive. It is noteworthy that the SSA-parallelization scheme changes the algorithmic behavior of the given MOEA to that of an asynchronous $(\mu + 1)$-evolutionary strategy.

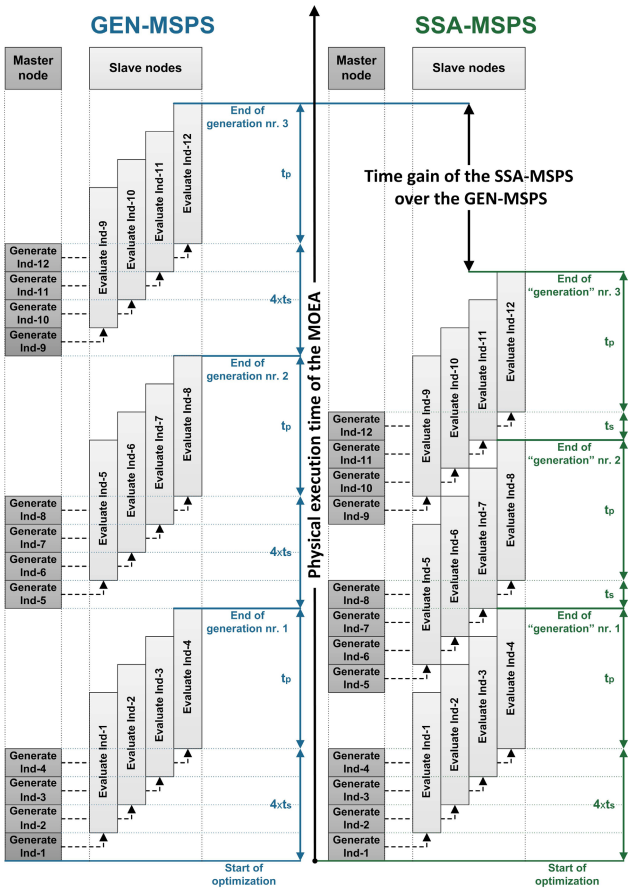


Fig. 2. The comparative computation steps of GEN-MSPS and SSA-MSPS for 3 generations of size 4 in a distributed computing environment with one master node and 4 slave nodes

Figure 2 provides an example of how individuals are processed by the two parallelization methods. The lack of a synchronization step in SSA-MSPS enables this

method to evaluate more individuals per time interval than GEN-MSPS. We shall investigate this quantitative aspect in Section 3. The negative side to using SSA-MSPS is that the same lack of generational synchronization is expected to make SSA-MSPS achieve worse results than GEN-MSPS in terms of Pareto front quality after evaluating a fixed number of individuals. This qualitative aspect is treated in Section 4. The last part of Section 4 contains an interpretation of the interplay between the quantitative and qualitative observations.

3 Examining the Quantitative Performance

3.1 The Basic Model

As mentioned in the previous section, it makes sense to presume that, given the same hardware architecture and specific MOEA settings, SSA-MSPS is able to compute faster than GEN-MSPS a given number of individuals (please see Figure 2). Another way of looking at this is that, by using the SSA-MSPS, one will be able to create and evaluate more individuals in the same time interval. We shall now attempt to quantify this improvement and to evaluate how the interplay between the duration of the parallel fitness evaluations and the duration of the synchronous tasks affects it.

Our theoretical model consists of a $(\mu + \lambda)$-archiving MOEA that is distributed over a computing environment with λ slave nodes. We mark with $t_p > 0$ the duration (in time units) of distributing and performing the fitness evaluation of any individual on any slave node. We also mark with $t_s > 0$ the cumulative duration of the sequential computation tasks (i.e., genetic operations + possible pre-evaluation tasks) that are performed on the master node in order to create one individual. In this section we assume that t_s and t_p are constant. We define the *parallelization ratio* as:

$$r = \lceil \frac{t_p}{t_s} \rceil \tag{1}$$

When considering the GEN-MSPS approach, it is quite straightforward that when it is using more than $r+1$ slave nodes simultaneously, the computation time is not further improved. As such, the following reasoning is made under the restriction $r+1 \geq \lambda$.

Assuming that other miscellaneous computation times are negligible with regards to (or integrated in) t_s and t_p, the total time required to compute any generation of λ individuals using the GEN-MSPS is $(\lambda \times t_s) + t_p$. In case of the SSA-MSPS, the time required to compute the first λ individuals is also $(\lambda \times t_s) + t_p$, but the time required to compute any of the next batches of λ individuals is $(t_s + t_p)$, as sketched in Figure 2.

As such, under the assumptions of our theoretical model, when wishing to compute N generations, the overall computation time is *a*) $(\lambda \times t_s + t_p) \times N$ in case of the GEN-MSPS scheme; *b*) $(\lambda \times t_s + t_p) + (t_s + t_p) \times (N-1)$ in case of the SSA-MSPS scheme. After equalizing these computation times and performing the necessary calculations, we obtain that in the time interval required by the GEN-parallelization to compute N generations of λ individuals, the SSA-parallelization can compute $\Delta_{struct}\%$ more individuals, where Δ_{struct} is given by:

$$\Delta_{struct} = \frac{(N-1)(\lambda-1) \times t_s}{N \times (t_s + t_p)} \times 100 \tag{2}$$

We shall refer to this measure as the **structural improvement** that SSA-MSPS has over GEN-MSPS in terms of computed individuals per given time interval.

It is important to note that while Δ_{struct} does depend on the number of generations, the population size and also on the order of size between these two variables and t_s, the dominant factor that influences Δ_{struct} is the ratio between t_s and t_p, or in other words, the parallelization ratio r. As we fix $\lambda = 100$, $N = 500$, and $t_s = 1$, by varying the value of t_p, we obtain the dependency of Δ_{struct} on r. The plot is presented in Figure 3 - basic model curve. Unsurprisingly, it shows that the quantitative improvement that SSA-MSPS brings, decreases exponentially with regards to the parallelization ratio.

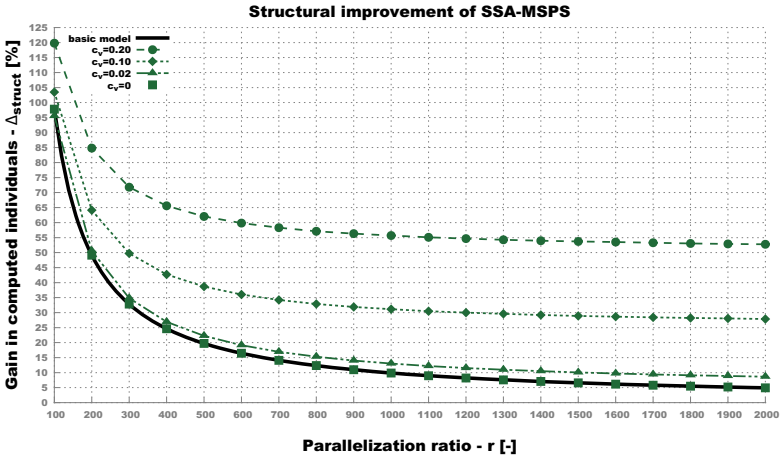


Fig. 3. Δ_{struct} curves for different parallelization ratios and different degrees of variance (i.e. c_v) in the time-wise distribution of the fitness evaluation function

Although valuable in establishing a baseline for the comparison between the GEN-parallelization and the SSA-parallelization schemes, the above described comparison has one severe limitation: it is strongly influenced by the idealistic assumption that the duration of the fitness evaluation task is constant (i.e. there is zero variance in the time-wise distribution of the fitness evaluation function). In the next subsection, we proceed to address this issue in order to enhance our quantitative performance model.

3.2 The Effect of Variance on the Quantitative Performance

Initially, we tested the theoretical model proposed in the previous section. Using a constant (i.e. variance free) time-wise fitness distribution, we simulated the time required by GEN-MSPS and SSA-MSPS runs when having various values of the coefficient of parallelization (1). The obtained results (Figure 3 - the $c_v = 0$ data points) confirm the Δ_{struct} behavior indicated by the theoretical model from (2).

Next, we evaluated the influence of having various degrees of variance in the time-wise distribution. In these tests, the fitness evaluation of each individual took t_p milliseconds where $t_p \sim \mathcal{N}(m, \sigma)$. As we fixed $t_s = 1$, in this case, $r \sim m$ and, when scaling up

m we also modified σ in order to keep the coefficient of variation, $c_v = \frac{\sigma}{m}$, constant at a preset value. The maximum amount of variance that we could consider, under a normal distribution assumption, was given by $c_v = 0.2$. Because of the induced stochasticity, for each value of r we performed 100 tests and we report averaged results.

The plot in Figure 3 shows how Δ_{struct} behaves when using four different c_v values. The curves clearly show that the logarithmic decrease of Δ_{struct} is less intense with increased variance. Further experiments have also shown that, with variance in the time-wise fitness distribution function, after reaching a lower threshold, the value of Δ_{struct} tends to stabilize. We have run simulations up to $r = 10^7$ with a step size of 10000 and, in Table 1, we report the discovered lower thresholds of Δ_{struct} for different variance levels. We mention that, in the absence of variance, for $r = 10^7$, $\Delta_{struct} = 0.000988\%$.

The conclusion of the above tests is that the theoretical model (2) gives an accurate lower limit for Δ_{struct} but the value of Δ_{struct} for a given parallelization ratio r is significantly higher when having variance in the time-wise fitness distribution. Furthermore, in the presence of variance, Δ_{struct} is lower bounded by variance-specific thresholds that display a remarkable stability even at very high values of r.

Table 1. The observed variance-specific lower thresholds of Δ_{struct}

c_v [-]	Lower threshold for Δ_{struct} [%]	Δ_{struct} for $r = 10^7$ [%]
0.20	50.0822	50.2134
0.10	25.0858	25.1224
0.05	12.5940	12.6135
0.02	5.0969	5.1017

4 Examining the Qualitative Performance – Empirical Results

4.1 Evaluation Framework Setup

The qualitative performance of the two considered master-slave parallelization schemes depends on the concrete MOOP to be solved, on the used MOEA and on the parameterization of the algorithm. In the following paragraphs we describe the details of the performance evaluation framework we used.

Test Problems. We have chosen for benchmarking purposes four standard, artificial, test problems from evolutionary multi-objective literature. Our choice of artificial problems is self-evident as it is very helpful to know the ground truth (i.e., the true Pareto front) in order to compare between parallelization results. The problems have been specially selected as they propose different degrees of difficulty and different convergence behaviors for the two MOEA algorithms that we experiment with. The MOOPs we consider are *a*) **KSW** - a classic optimization problem with 10 variables and two objectives based on Kursawe's function [11]; *b*) **DTLZ7** - a problem with 22 variables and 3 objectives that is aimed at testing the performance of a MOEA on discontinuous

Pareto fronts [5]; *c*) **ZDT6** - a problem with 10 variables and 2 objectives that proposes difficulties regarding the non-uniformity of the search space [14]; *d*) **LZ09-F1** - a problem with 30 variables and two objectives part of the LZ09 problem set [12] which is particularly difficult for classic MOEAs like NSGA-II and SPEA2.

The computation of the fitness values for all four problems is very fast on any modern processor. In order to make the MOEAs exhibit the desired test behavior, the fitness computation times were artificially increased (as described in Section 3.2).

MOEA Parameterization. Our implementation of the NSGA-II and SPEA2 algorithms is loosely based on the one provided by the jMetal package [7]. Because our main goal is to compare the performance of two different parallelization models on the same algorithm, we use, more or less, standard parameterization options.

As such, for all the performed tests, we used a population size of 100 individuals and an archive size of 100. In the case of SSA-MSPS the term "generation" is used to denote a batch of 100 individuals. We used three standard genetic operators from MOEA literature: binary tournament selection, simulated binary crossover [3] and polynomial mutation. We used standard values to parameterize these genetic operators: 0.9 for the crossover probability, 20 for the crossover distribution index, $1/L$ for the mutation probability (where L is the number of variables) and 20 for the mutation distribution index.

We make our comparisons based on rather short runs of 500 generations (50.000 individuals). This is because after 500 generations we reach generally good solutions for all the considered test problems and because we are particularly interested in studying the middle-stage convergence behavior of GEN-MSPS and SSA-MSPS. We consider that the behavior of the two parallelization schemes in the late-stage of convergence is of less practical importance for us. This is because time constraints are very important in many real-world industrial optimization scenarios and practitioners rarely run an optimization for more than a few hundred generations. If one particular method is constantly outperforming the other in a late-stage of convergence, given the descriptions from Section 2, one can easily switch during the optimization to the best performing parallelization method by enabling or disabling the mentioned **synchronization step**.

Quality Indicators. For a given solution set S, the hypervolume associated with this solution set, $\mathscr{H}(S)$, has the advantage that it is the only Pareto front quality estimation metric for which there is a theoretical proof [9] of a monotonic behavior. This means that the maximization of the hypervolume constitutes the necessary and sufficient condition for the set of solutions to be *maximally diverse Pareto optimal solutions of a discrete, multi-objective, optimization problem* [9]. In light of this, for any optimization problem, the *true Pareto front* has the highest achievable hypervolume value.

In our case, for a given MOOP, the monotonic property of the hypervolume metric makes it ideal for assessing the relative quality of an arbitrary solution set S_*. Let us mark with S_{true} the *true Pareto front* of our MOOP. As we deal with artificial problems where S_{true} is known, we can present the quality of the given solution set as a percentile obtained by reporting the hypervolume measure of this solution set to the hypervolume value associated with the true Pareto front of the given MOOP:

$$qual(S_*) = \frac{\mathscr{H}(S_*)}{\mathscr{H}(S_{true})} \times 100 \tag{3}$$

Expressing the quality of a solution set as a true hypervolume percent also enables us to define more accurately what we mean by the syntagms early, middle and late-stage of convergence. For the purpose of this research, in light of the motivations presented in the last paragraph of the previous section, we define a MOEA as being in the early stage of convergence if $qual(A_*) \leq 15$ where A_* denotes the current archive of the MOEA. If $qual(A_*) \in (15, 85]$ we consider the algorithm to be in the middle-stage of convergence, while $qual(A_*) > 85$ is associated with a late-stage of convergence.

Consider we wish to solve our MOOP with a certain MOEA. Let: *a)* $p = \overline{1,100}$ be an integer value; *b)* $C_{GEN}(p)$ be the minimal number of individuals that must be computed when using GEN-MSPS in order to reach a solution set S_1 with the property that $qual(S_1) \geq p$; *c)* $C_{SSA}(p)$ be the minimal number of individuals that must be computed when using SSA-MSPS in order to reach a solution set S_2 with the property that $qual(S_2) \geq p$. For our MOOP-MOEA combination, we define the *SSA qualitative deficit* at the true hypervolume percentile p as:

$$\Delta_{qual}(p) = \left(\frac{C_{SSA}(p)}{C_{GEN}(p)} - 1 \right) \times 100 \tag{4}$$

This $\mathscr{H}$-derived measure shows the difference in the minimum number of individuals that must be computed when using the two parallelization schemes in order to reach a solution set S_p with the property that $qual(S_p) \geq p$.

In each series of tests that we have performed, we made 50 runs for each parameter configuration and we always report over averaged results. The authors are aware that a balanced comparison of non-dominated solution sets often takes into consideration more than a single performance metric [16], but we consider that, on its own, the hypervolume metric is sufficient for expressing major qualitative differences between solutions sets, especially when reporting over averaged results.

4.2 Basic Qualitative Performance Tests

In the first series of performed tests, using a constant fitness distribution (i.e. $c_v = 0$), we measured the quality of the MOEA archive (in terms of hyper-volume) after the completion of each generation (i.e. batch of 100 individuals in the case of SSA-MSPS). These results are presented in the subplots in Figure 4 marked with (a). We consider that a more useful perspective for presenting the same comparative convergence behavior information can be constructed by plotting the Δ_{qual} values of the MOEA archive (subplots marked with (b) from Figure 4).

The results of these initial tests confirm some of the findings in [6], in the sense that the GEN-MSPS is able to achieve a higher quality Pareto front than SSA-MSPS after the same number of evolved individuals in the early and middle stages of convergence for all the considered problems. Furthermore, when abstracting the behavioral shifts that characterize the early and late stages of convergence, we notice that Δ_{qual} values are quite constant for each test problem. By averaging the individual Δ_{qual} values associated

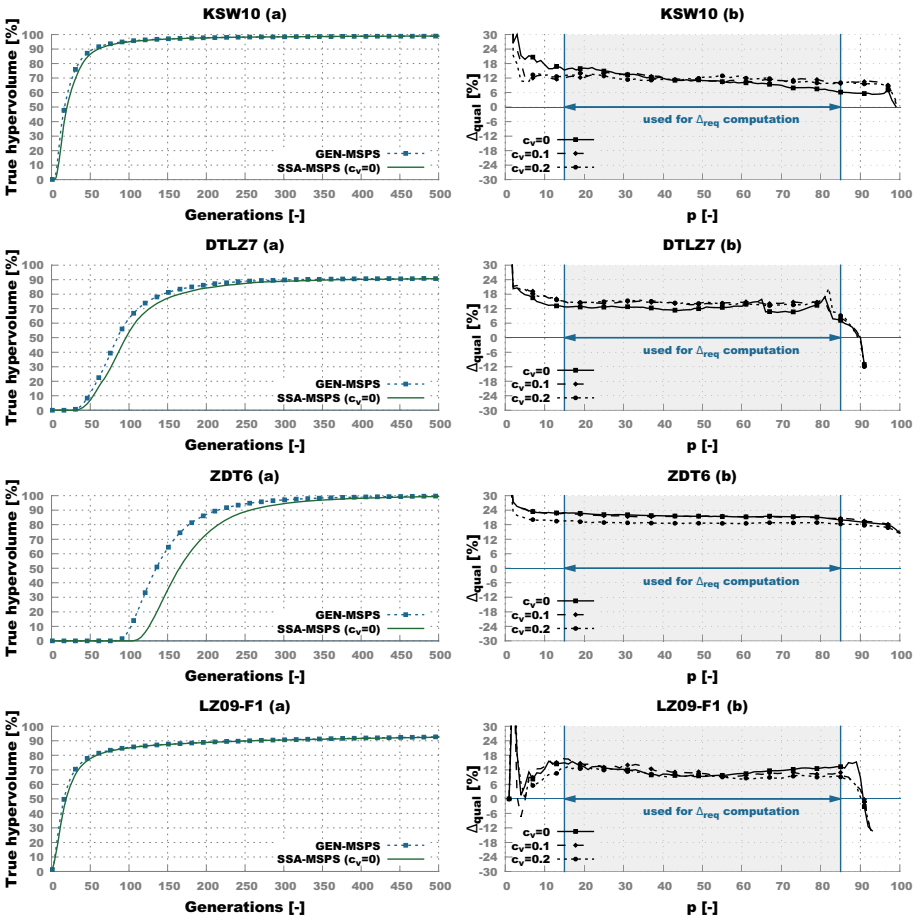


Fig. 4. SPEA2 generation-wise qualitative performance plots averaged over 50 runs

with the middle stage of convergence (i.e., $p = \overline{16,85}$), we obtain the **average required SSA improvement** for each MOOP-MOEA combination:

$$\Delta_{req} = \frac{1}{70} * \sum_{p=16}^{85} \Delta_{qual}(p) \tag{5}$$

4.3 The Effect of Variance on the Qualitative Performance

The second series of tests that we have performed in order to gain more insight into the qualitative performance of the two parallelization schemes is again related to the influence of having variance in the time-wise fitness distribution function.

The results obtained using SPEA2 are presented in the subplots marked with (b) from Figure 4. The values of Δ_{req} for both SPEA2 and NSGA-II are shown in Table 2. It is easy to observe that, in the case of the qualitative performance, variance in the time-wise distribution of the fitness evaluation function has a negligible effect: Δ_{req} is not

directly proportional to the amount of variance and, in half of the cases, the observed average changes induced on Δ_{req} by having some level of variance are not statistically significant. This creates a stark contrast when comparing with the effect that variance has on the quantitative performance (Section 3.2) and provides a solid indicator that SSA-MSPS should be favored in the presence of significant variance.

Table 2. Averaged values of the Δ_{req} metric over 50 runs for different levels of variance in the time-wise distribution of the fitness evaluation function. For each MOEA - MOOP combination, the highest value is highlighted and marked with "$^+$" if the difference between it and the lowest Δ_{req} value of the combination is statistically significant (one-sided Mann-Whitney-Wilcoxon test with a considered significance level of 0.05).

	Δ_{req} for **SPEA2** [%]				Δ_{req} for **NSGA-II** [%]			
Problem	$c_v = 0$	$c_v = 0.02$	$c_v = 0.10$	$c_v = 0.20$	$c_v = 0$	$c_v = 0.02$	$c_v = 0.10$	$c_v = 0.20$
KSW10	11.06	11.59	11.77	11.64	12.65	11.41	12.25	12.62
DTLZ7	12.13	16.16^+	14.10	14.21	18.46	20.43^+	16.58	17.79
ZDT6	21.55^+	20.22	21.39	18.75	21.91	22.33	22.67	23.32
LZ09-F1	11.49	14.02^+	11.47	10.04	12.71	13.73	12.86	14.04

4.4 The Interplay between the Quantitative and the Qualitative Aspects

This stable behavior of the qualitative deficit exhibited by SSA-MSPS allows for a simple reasoning regarding the comparative performance of GEN-MSPS and SSA-MSPS: *if, for a given optimization setup, the quantitative improvement of SSA-MSPS (Δ_{struct}) can overcompensate the qualitative deficit of SSA-MSPS (Δ_{req}), we can say that, on average, SSA-MSPS is the better parallelization choice.* This is because when ($\Delta_{struct} > \Delta_{req}$) we have good reasons to believe that, in any middle-stage convergence time interval, SSA-MSPS can produce Pareto fronts that are better or at least as good as those that might have been obtained by using GEN-MSPS for the same time interval.

Graphically, we can combine the quantitative performance and the qualitative performance by simply plotting Δ_{req} as a constant value (please see Figure 5). The intersection between the Δ_{req} constant line and the Δ_{struct} curve is the problem specific *parallelization equilibrium point*. If the equilibrium point lies to the right of the parallelization ratio, on average, SSA-MSPS is the better option, while if the equilibrium point lies to the left of the parallelization ratio, GEN-MSPS should be preferred. Figure 5 contains two charts that show the best parallelization decisions for our four test problems, under the assumptions of a constant time-wise distribution of the fitness function and of a parallelization ratio $r = 650$. SSA-MSPS is the better parallelization choice for KSW10, DTLZ7 and LZ09-F1 when using SPEA2. In the case of NSGA-II, SSA-MSPS performs better only in the case of KSW10 and LZ09-F1.

Applying this quantitative and qualitative analysis of parallelization performance in real world optimization scenarios is rather straightforward. Nevertheless, the posteriori character of both Δ_{struct} and Δ_{req} means that a few initial, short, mock-up test runs are required in order to estimate the concrete parallelization ratio and the problem specific qualitative behavior. This short profiling phase is more than worthwhile when one has to solve multiple MOOPs that fall within the same class (e.g., problems with roughly

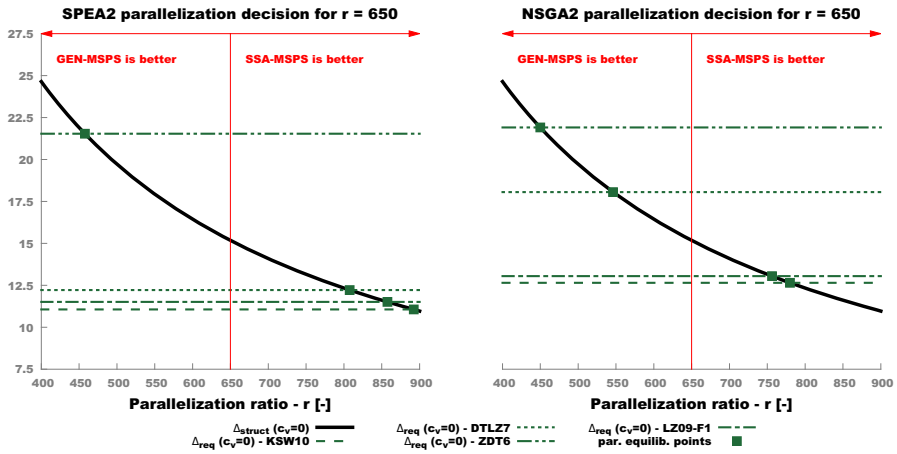


Fig. 5. Graphical example of the best parallelization decisions for specific MOOP-MOEA combinations. The hypothetical parallelization ratio is set at $r = 650$.

similar parameter vectors and / or different parameter ranges). For the same MOOP class, Δ_{req} is expected to be quite robust while Δ_{struct} can be generally estimated quite quickly as it is solely dependent on the concrete parallelization setup.

5 Conclusions and Future Work

In this paper, we investigated two master-slave parallelization methods (generational and steady state asynchronous) for MOEAs and tried to discover what are the decisive factors that can make one method outperform the other in a time-constrained optimization scenario that also requires very time-intensive fitness evaluation functions.

In order to achieve this, we performed a comparative quantitative and qualitative analysis of the behavior of these two methods when applying them to parallelize NSGA-II and SPEA2 optimization runs. The results indicate that 1) *the parallelization ratio* and especially 2) *the level of variance in the time-wise distribution of the fitness evaluation function* are the key factors that influence the relative performance of the two parallelization methods. The presence of variance is a key factor, as a rather heterogeneous fitness function can make the steady state asynchronous parallelization method (SSA-MSPS) considerably outperform its generational counterpart (GEN-MSPS).

In the future, we plan to profile using Δ_{req} more problems, including industry proposed MOOPs, and we want to test the parallelization schemes on more modern MOEAs.

References

1. Coello, C., Lamont, G., Van Veldhuisen, D.: Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation Series. Springer (2007)
2. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester (2001)

3. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Systems 9, 115–148 (1995)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
5. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: IEEE Congress on Evolutionary Computation (CEC 2002), pp. 825–830 (2002)
6. Durillo, J., Nebro, A., Luna, F., Alba, E.: On the effect of the steady-state selection scheme in multi-objective genetic algorithms. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) EMO 2009. LNCS, vol. 5467, pp. 183–197. Springer, Heidelberg (2009)
7. Durillo, J.J., Nebro, A.J.: JMETAL: A Java framework for multi-objective optimization. Advances in Engineering Software 42, 760–771 (2011)
8. Durillo, J., Nebro, A., Luna, F., Alba, E.: A study of master-slave approaches to parallelize NSGA-II. In: IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2008), pp. 1–8 (2008)
9. Fleischer, M.: The measure of Pareto optima. applications to multi-objective metaheuristics. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 519–533. Springer, Heidelberg (2003)
10. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. In: Foundations of Genetic Algorithms, pp. 69–93. Morgan Kaufmann (1991)
11. Kursawe, F.: A variant of evolution strategies for vector optimization. In: Schwefel, H.-P., Männer, R. (eds.) PPSN 1990. LNCS, vol. 496, pp. 193–197. Springer, Heidelberg (1991)
12. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation 13(2), 284–302 (2009)
13. Yagoubi, M., Thobois, L., Schoenauert, M.: Asynchronous evolutionary multi-objective algorithms with heterogeneous evaluation costs. In: IEEE Congress on Evolutionary Computation (CEC 2011), pp. 21–28 (2011)
14. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8(2), 173–195 (2000)
15. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou, K.C., et al. (eds.) Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001), pp. 95–100. International Center for Numerical Methods in Engineering, CIMNE (2002)
16. Zitzler, E., Thiele, L., Bader, J.: On set-based multiobjective optimization. IEEE Transactions on Evolutionary Computation 14(1), 58–79 (2010)
17. Zăvoianu, A.-C., Bramerdorfer, G., Lughofer, E., Silber, S., Amrhein, W., Klement, E.P.: A hybrid soft computing approach for optimizing design parameters of electrical drives. In: Snasel, V., Abraham, A., Corchado, E.S. (eds.) SOCO Models in Industrial & Environmental Appl. AISC, vol. 188, pp. 347–358. Springer, Heidelberg (2013)

TRCM: A Methodology for Temporal Analysis of Evolving Concepts in Twitter

Mariam Adedoyin-Olowe[1], Mohamed Medhat Gaber[1], and Frederic Stahl[2]

[1] School of Computing, University of Portsmouth
Hampshire, England, PO1 3HE, UK
[2] School of Systems Engineering, University of Reading
P.O. Box 225, Whiteknights, Reading, RG6 6AY, UK

Abstract. The Twitter network has been labelled the most commonly used microblogging application around today. With about 500 million estimated registered users as of June, 2012, Twitter has become a credible medium of sentiment/opinion expression. It is also a notable medium for information dissemination; including breaking news on diverse issues since it was launched in 2007. Many organisations, individuals and even government bodies follow activities on the network in order to obtain knowledge on how their audience reacts to tweets that affect them. We can use postings on Twitter (known as tweets) to analyse patterns associated with events by detecting the dynamics of the tweets. A common way of labelling a tweet is by including a number of hashtags that describe its contents. Association Rule Mining can find the likelihood of co-occurrence of hashtags. In this paper, we propose the use of temporal Association Rule Mining to detect rule dynamics, and consequently dynamics of tweets. We coined our methodology Transaction-based Rule Change Mining (TRCM). A number of patterns are identifiable in these rule dynamics including, new rules, emerging rules, unexpected rules and 'dead' rules. Also the linkage between the different types of rule dynamics is investigated experimentally in this paper.

Keywords: Twitter, Hashtags, Association Rule Mining, Association Rules, New Rules, Emerging Rules, Unexpected Rules, Dead Rules, Rule Matching.

1 Introduction

The surge in the acceptability of Twitter since its launch in 2007 has made it the most commonly used microblogging application [8,16] (in this paper we used the terms *Twitter*, *Twitter network* and *network* interchangeably). The network permits the effective collection of large data which gives rise to major computational challenges. More people are becoming interested in and are relying on Twitter for information and news on diverse topics. Twitter is mainly known for short instant messaging that allows a maximum of 140 characters per message (tweet). Users follow other users' comments or contributions on events taking place globally in real time [5]. The network is labelled the most commonly used

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 135–145, 2013.

microblogging application around today with about 500 million estimated registered users as of June, 2012 [11]. Twitter has become a strong medium of opinion expression and information dissemination on diverse issues [20]. It is also a remarkable source for breaking news broadcasting [3,14]. Considering the enormous volume of tweets generated on a daily basis, users have invented a common way of labelling tweets. This is often attained by including a number of hashtags as prefix to keywords in tweets to describe the tweet's contents. The use of hashtags makes it easy to search for and read tweets of interest.

Tweets posted online include news, major events and topics which could be of local, national or global interest. Different events/occurrences are tweeted in real time all around the world making the network generate data rapidly. The network reports useful information from different perspectives for better understanding [22]. Twitter as a social network and hashtags as tweet labels can be analysed in order to detect changes in event patterns using Association Rules (ARs).

In this paper we use Association Rule Mining (ARM) to analyse tweets on the same topic over consecutive time periods t and $t+1$. We also use Rule Matching (RM) to detected changes in patterns such as 'emerging', 'unexpected', 'new' and 'dead' rules in tweets. This is obtained by setting a user-defined Rule Matching Threshold (RMT) to match rules in tweets at time t with those in tweets at $t+1$ in order to ascertain rules that fall into the different patterns. We coined this proposed method Transaction-based Rule Change Mining *(TRCM)*. Finally, we linked all the detected rules to real life situations such as events and news reports.

This paper is organised as follows. Section 2 covers related work to our research. Methods for rule dynamics discovery including a background of ARM is provided in Section 3. Our proposed method *TRCM* is provided in Section 4. Section 5 provides an experimental case study. Potential applications of the proposed method are given in Section 6. The paper is concluded in Section 7 with a summary.

2 Related Work

The authors of [17,18,21] refer to Twitter as a medium of expressing sentiment and opinion on diverse issues. In the research of [4] the chatter from Twitter.com was used to forecast box-office revenue for movies. The work of [17] reveals that postings on Twitter can influence decision making of different entities. Twitter is also known as a strong medium for the broadcast of breaking news [15]. There may not be news agents around on the scene of incident but there will always be tweeters to broadcast the event live on Twitter even before traditional news agents appear on the scene. Twitter is as well a network for information dissemination and awareness creation on diverse topics [6,22].

Related work on Twitter has shown that the network is a viable platform for evaluating people's opinion. The work of [18] created classification models from sentiment expressed on a common disease in Brazil. On the other hand, [22] applied the empirically authenticated MEDIC approach to identify interesting hashtags. This approach was applied to draw the attention of targeted users on the

network based on follow links, tagged contacts and topographical /geographical fusion of users [20]. The research of [7] was based on influencer on the network. They employed Spearman's rank correlation coefficient to determine the strength of association between two rank sets. However, the experiment of [21] was based on a graph-based hashtag sentiment classification approach. Their major study was on sentiment polarity of collective hashtags within a specified time frame rather than individual hashtags polarity. The research of [20] was based on the formation of Twitter community and the aims of the different users that constitute the network. They used the HITS algorithm to propose a two-level structure for the user aim discovery and detection of core influencers in the network. Work on Twitter as word of mouth medium of advertisement was carried out by [10]. They used a case study approach on some branded products. The investigation of [4] was centred on box office revenues based on film reviews through Twitter using the LingPipe linguistic analysis package they called DynamicLMClassifier. The authors of [5] offered multinomial Naive Bayes, stochastic gradient descent and Hoeffding tree which are efficient in dealing with data streams to discover sentiment knowledge in Twitter. They also proposed sliding window Kappa statistic for assessment in time limited data streams. Furthermore, the work of [6] investigated the effect of public mood portrayed in tweets on the stock market. We identified the work of [12] as closely related to our work. However, they proposed Event Change Detection (ECD) to show how changes in event trends can be used for decision support in environment scanning. Our work used ARs to detect changes in hashtag trends based on time drift and event trends on Twitter. They discussed emerging event patterns which signify rules with momentous increase pattern. We applied ARs to discover hashtags that became popular as a result of sudden occurrence (for example breaking news), thereby shifting the attention of tweeters to another event altogether (emerging rules).

3 Rule Dynamics of Association Rule Mining

Association Rule Mining (ARM) is a data mining technique made popular by Agrawal et al [1], when they came up with the Apriori algorithm for the discovery of frequent itemsets and strong association rules. ARM is an act of extracting interesting recurrent representation, associations or links, between different arrays of items within transactional databases (Market Basket), relational databases (Personal Details), or any other information warehouses [12,13] in the form of rules. It discovers and reveals remarkable associations embedded in huge data sets which may include hidden information that can be useful for decision making [9]. This technique tends to reveal every probable association that satisfies definite boundaries using the lowest support and confidence [2]. An association rule is in the form $X => Y$, where X and Y are disjoint sets of items. In this paper, we applied ARM to hashtags in tweets to discover trends in dynamic rules and identify when different rules appear and disappear on Twitter based on event changes.

We used the left hand side (lhs)/conditional and the right hand side (rhs)/ consequent parts of rules in Apriori principle to analyse hashtags as conveyed

Table 1. The Tweet Matrix

Tweet 1	#datamining	#bigdata	#sql	#KDD
Tweet 2	#ecommerce	#ISMB	#datamining	
Tweet 3	#bigdata	#facebook	#data mining	#analytics
Tweet 4	#analytics	#privacy	#datamining	
Tweet 5	#datamining	#KDD	#bigdata	

in tweets over a period of time. The analysis of these hashtags is used to detect Association Rules (ARs) present in the tweets at different points in time. The similarities and differences in the ARs in the tweets at time t and time $t+1$ are measured to reveal the 'emerging', 'unexpected', 'new', and 'dead' rules in tweets as they evolve from being strong rules to weak and then 'dead'.

The capability of the ARM technique enables it to uncover different patterns in both transactional and relational datasets. Changes in rules dynamics patterns generated using the Apriori principle can be used as a decision support tool. An example of this is breaking news of a disaster, say an earthquake in Japan. The news will generate high rules in tweets at the early stage. This is referred to as speedy rule emergence. The emergence of this rule can result in rapid broadcast of breaking news by news agencies all around the world. It can also help other organisations like the 'Red Cross' to respond swiftly and dispatch aids to the affected areas more quickly.

3.1 Notation

Both [12] and [19] came up with methods for calculating similarities and differences between two rules at different points in time. Given that the methods adopted in [12] and [19] were designed for association rules discovered from relational datasets (general association rules). Our proposed methodology is this paper defines the similarity based on the concepts of degree of similarity proposed in [12] and [19]. Details of the calculations and notation used are given in the following.

n	Number of hashtags
i	an association rule in dataset 1 presented in binary vector
j	an association rule in dataset 2 presented in binary vector
lh_i/lh_j	number of hashtags with value 1 in conditional part of rule i/j
rh_i/rh_j	number of hashtags with value 1 in consequent part of rule i/j
lh_{ij}/rh_{ij}	number of same hashtags in conditional/consequent part of rules i and j
p_{ij}/q_{ij}	degree of similarity of features in conditional/consequent part of rules i and j
r_j^t	Rule present at time t
r_j^{t+1}	Rule present at time $t+1$

3.2 Measuring Similarity

$$p_{ij} = \frac{lh_{ij}}{max(lh_i, lh_j)} \tag{1}$$

$$q_{ij} = \frac{rh_{ij}}{max(rh_i, rh_j)} \tag{2}$$

Equations 1 and 2 p_{ij} and q_{ij} were adopted from [12]. They show the similarity in the conditional and consequent parts of rule i and rule j at different points in time t and $t+1$ in that order. Change in rules can be discovered by matching every rule in ruleset t with those in ruleset $t+1$ using the similarity and difference comparison.

3.3 Rule Matching

The authors of [19] used ARM to mine the change in customer behaviour in an Internet shopping mall. They proposed RM as the most common approach to discovering rule changes in two datasets. In this paper, we used RM to detect change in patterns of rules in r_j^t and r_j^{t+1}, where t is the time and j is the rules present in the tweets. Rules in r_j^{t+1} are matched against rules in r_j^t to detect patterns of rule changes.

The RM threshold in our experiment is set in the interval [0, 1], where 0 indicates dissimilarity and 1 indicates similarity. Please note that all rules are considered as 'new' until there is a matching rule found.

3.4 Patterns in Tweet Change Discovery

Having explained RM, we will now discuss Tweet Change Discovery (TCD) which is an integral part of our research. The TCD is used to develop our Transactional-based Rule Change Mining (TRCM) framework. In TRCM hashtags in the conditional and consequent parts of the rules are observed, as presented in Fig. 1, to detect the exact rule change pattern that has occurred. A framework showing the process of Tweet Change Discovery and our TRCM framework are presented in Fig. 1 and Fig. 2 respectively.

The five patterns in TCD identified in this research are defined and explained as follows:

Unexpected Consequent Change in Tweet. Where a rule in r_j^t and r_j^{t+1} has similar conditional part but different consequent part ($p_{ij} \geq thp_{ij}$ and $q_ij < thq_{ij}$), then there is an unexpected consequent change provided the difference measure between r_j^t and rules r_j^{t+1} is greater than 0 (say 0.62). However, the status of this rule changes when there is a similar event pattern in $t+1$ that has a higher similarity measure (for example 0.90) than the initial difference detection. In this case an emerging change is said to have occurred. The implication of these two scenarios is that emerging change overwrites unexpected change.

Unexpected Conditional Change in Tweet. An unexpected conditional change is detected when the consequent parts of rule r_j^t at and r_j^{t+1} are similar,

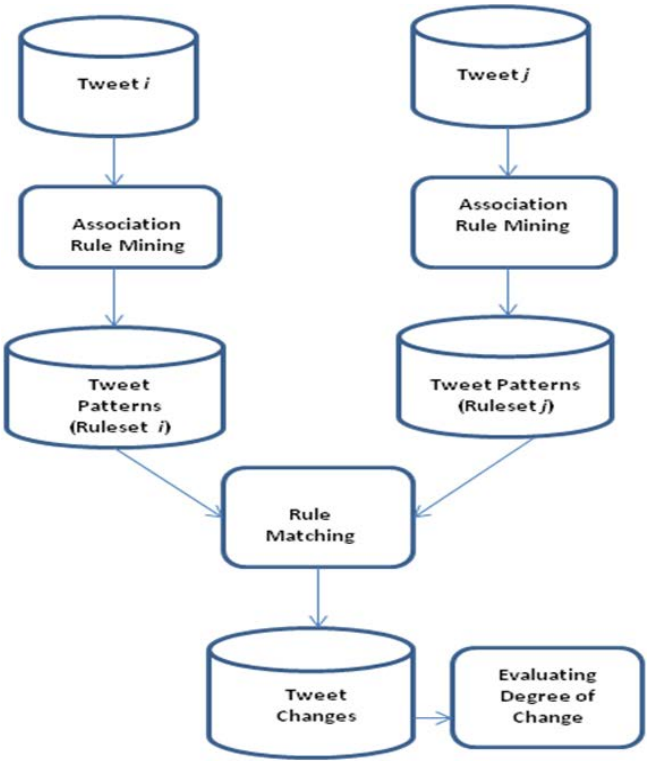


Fig. 1. The Process of Tweet Change Discovery

but the conditional parts are different ($p_{ij} < thp_{ij}$ and $q_{ij} \geq thq_{ij}$). If the absolute difference measure is less than 0, then the consequent part is similar and the conditional part is different. On the other hand, if the absolute value of the difference measure is greater than 0, then an unexpected conditional change is detected in the tweet.

Emerging Change in Tweet. Emerging rules are discovered when two hashtags at time t and $t+1$ have similarities greater than the user-defined threshold ($p_{ij} \geq thp_{ij}$ and $q_{ij} > thq_{ij}$). The similarities take place in both the conditional and consequent part of the rule.

New Rules. All rules are said to be 'new' until there is a matching rule found. Every hashtag at time $t+1$ is completely different from all the hashtags in time t ($p_{ij} < thp_{ij}$ and $q_{ij} < thq_{ij}$).

'Dead' Rules. A 'Dead' rule occurrence is the opposite of new rules detection. A rule in t is labeled 'dead' if its maximum similarity measure with all the rules in $t+1$ is less than the user-defined threshold ($p_{ij} < thp_{ij}$ and $q_{ij} < thq_{ij}$). Fig.2 (rulesets t and $t+1$) shows the different rules patterns already discussed.

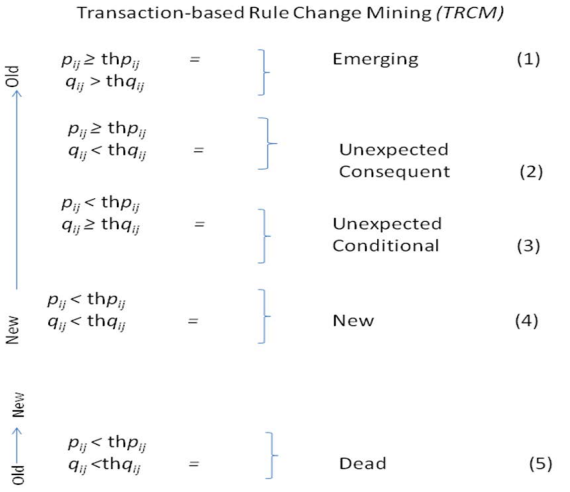


Fig. 2. Transaction-base Rule Change Mining (TRCM)

4 Transaction-base Rule Change Mining (TRCM)

TRCM is our research framework that defines rule change patterns in tweets at different period of time. In 1 - 4 hashtags in new rules (r_j^{t+1}) are matched with those in old rules (r_j^t).

On the other hand, in 5 hashtags in old rules (r_j^t) are matched with hashtags in rules r_j^{t+1} as shown in Fig.2. In order to measure the TCD we propose the following steps.

Step 1: For each rule in the conditional part of the new ruleset r_j^{t+1}, match with all rules in the conditional part of the old ruleset r_j^t. Compute the number of hashtags that appear in the conditional parts of both rulesets. In this case we have #KDD, #excel appearing in r_j^t and r_j^{t+1}.

Step 2: Divide the computed figure in step 1 (2, #KDD, #excel) by the maximum number of hashtags with value 1 in the conditional parts of either the old or new ruleset. For example in the conditional part of ruleset r_j^{t+1} (new rules), 5 hashtags (#KNN, #sqlserver, #analytic, #Facebook and #Privacy) have value of 1 each. However in ruleset r_j^t (old rules) have 2 hashtags with value 1 (#DT and #CART). Since r_j^{t+1} have a higher number of 5, then:

$$p_{ij} = \frac{lh_{ij}}{max(lh_i, lh_j)} = \frac{2}{5} = 0.4$$

lhs	rhs	supp	conf	lift
1{DT} =>	{#datamining}	1.0	1.000	1
2{CART} =>	{#DT}	1.0	1.000	1
3{#excel} =>	{#datamining}	0.4	1.000	1
4{#excel} =>	{#KDD}	0.4	0.625	1
5{#sql} =>	{#datamining}	0.4	0.435	1
6{#sql} =>	{#KDD}	0.4	0.555	1
7{#datamining} =>	{#KDD}	1.0	0.422	1
8{#KDD} =>	{#datamining}	1.0	1.000	1
9 {#datamining, #excel} =>	{#KDD}	0.4	1.000	1
10 {#excel, #KDD} =>	#datamining}	0.4	1.000	1
11 {#datamining, #sql} =>	{#Datamining}	0.4	0.112	1
12 {#KDD,#sql} =>	#datamining}	0.4	0.122	1

Fig. 3. Transaction-base Rule Change Mining (TRCM)

Step 3: Apply the same method in step 2 to the consequent parts of the two rulesets to detect the q_{ij}.

$$q_{ij} = \frac{rh_{ij}}{max(rh_i, rh_j)} = \frac{2}{3} = 0.6$$

Step 4: Identify the degree of similarity of rules in the old and new rules. However, for two rules to be similar, their degree of similarity must be greater than the pre-defined RMT which is 1. Where the degree of similarity is less than the RMT, the rules are said to be different. With the foregoing, the computation of the p_{ij} and q_{ij} in our sample datasets shows that neither the conditional nor the consequent parts of the rules are similar. This is because the p_{ij} (0.4) and the q_{ij} (0.66) are lower than the pre-defined RMT of 1.

5 Experimental Case Study

We applied the Apriori method on hashtags in rules r_j^t and r_j^{t+1} to mine the rule changes and detect the change pattern over time. This was obtained by declaring all the rules in the two datasets (Fig.3 and 4) as new rules. Rules in r_j^{t+1} were then matched one after the other with rules r_j^t. If there is a matching in rules, then the RMT is checked to determine the type of rule change found. All the rules in r_j^{t+1} were matched against those in r_j^t. Finally any rule in r_j^t that was still left unmatched is labeled 'dead' as the rule no longer exist.

6 Applications of TRCM to the Real World

Twitter is a dynamic network that produces several hundreds of millions of tweets every day. These data are often related to different events and event changes in the real world. Our proposed TRCM methodology is applied to tweets and these facts were discovered:

lhs	rhs	support	confidence	lift
1	{#KNN} => {#datamining}	1.0	0.111	1
2	{#sqlserver} => {#bigdata}	0.1	0.113	1
3	{#excel} => {#datamining}	1.1	0.110	1
4	{#KDD} => {#sqlserver}	0.1	1.000	1
5	{#excel} => {#datamining}	0.1	0.116	1
6	{#bigdata} => {#KDD}	0.1	0.111	1
7	{#analytic} => {#datamining}	0.1	0.110	1
8	{{#KDD} => {#datamining}	1.1	1.000	1
9	{#bigdata} => {#excel}	0.1	1.000	1
10	{#Facebook} => {#datamining}	0.1	0.111	1
11	{#privacy} => {#datamining}	0.1	0.114	1
12	{#bigdata} => {#excel}	0.1	0.117	1

Fig. 4. Transaction-base Rule Change Mining (TRCM)

- Different entities such as business organisations, personalities and governments can use the TRCM architecture to discover change trends in tweets. The management of business organisations can adjust any aspect of their policy/decisions that does not receive a 'thump up' on the Twitter network. Real life examples are cases of the Apple iTunes icon change, GAP clothing company logo change and MySpace logo change (all within the year 2012). These changes are put forward by their organisations but received a high degree of criticism from the social media with Twitter network in the forefront. Using TRCM in future, these organisations will be able to detect early the (negative) reactions of their customers and make a prompt adjustment to their decision making. By doing so, detrimental changes are guarded. TRCM can therefore serve as a decision support tool for different entities that deal with people.
- News agents can also use TRCM to discover news trends and report news that arises from among other news as emerging news (news that the similarity degree is greater that the user-defined threshold). An example is the breaking news of an earthquake in Japan described earlier in this paper. Not only will this news come across as an emerging rule, but also as a speedy rule emergence which will need to be reported immediately.
- Similarly TRCM is also useful to Twitter users, in that it will aid their activities on the network, which may be in the form of contributions, retweets and following topics on the network. Emerging trends can be captured and re-tweeted in order to increase its chances of being read and consequently draw the attention to the necessary audience of the topic.

The results of our research show the capability of TRCM of being a reliable tool in the effectiveness and beneficial use of Twitter network. More investigations will be conducted to discover how the degree of support and confidence can be used to reveal the rate of speed in rule change. Investigation will also be made to detect at what point different rules appear on the network and when they finally disappear. This experiment will propose a robust architectural framework that will be able to automatically detect all these changes.

7 Conclusion

Twitter data has become important to different entities in so many ways. To individuals, it has become a tool for awareness creation on diverse issues. It has also become a medium of information dissemination, which includes breaking news. Individuals sometimes rely on the Twitter network to obtain opinion of others while deciding on items to purchase, film to watch at the cinema or even political candidate to vote for during elections. The enormous data generated on the network need data mining techniques such as ARM to analyse and classify tweets, to detect their similarities and differences in relation to event patterns. The rule dynamics of ARM shows that tweets go through different phases. We proposed the Transaction-based Rule Change Mining (TRCM) that detects rules changes based on hashtags present in tweets and how the changes relates to events/occurrences in the real world. All the rules detected can be applied to the real world as a decision support tool for different entities, including individuals, organisations and government.

References

1. Agrawal, R., Imielinski., S.A.: Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (1993)
2. Ale, J., Rossi, G.: An Approach to Discovering Temporal Association Rules. In: Proc. of the 2000 ACM SAC, vol. 1, pp. 294–300 (2000)
3. Alfred, H.: Twittering the news: The Emergence of Ambient Journalism. Social Science Research Network, Journalism Practice 4(3), 297–308 (2010)
4. Asur, S., Huberman, B.: Predicting the future with social media. arXiv:1003.5699 [cs.CY] (2010)
5. Bifet, A., Frank, E.: Sentiment Knowledge Discovery in Twitter Streaming Data. In: Pfahringer, B., Holmes, G., Hoffmann, A. (eds.) DS 2010. LNCS, vol. 6332, pp. 1–15. Springer, Heidelberg (2010)
6. Bollen, J., Mao, H., Pepe, A.: Modelling public mood and emotion: twitter sentiment and socio-economic phenomena. In: Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (2009)
7. Cha, M., Haddadi, H., Benevenuto, F., Gummadi, K.: Measuring User Influence in Twitter: The Million Follower Fallacy. In: Proceedings of the 4th International AAAI Conference on Weblogs and Social Media, pp. 10–17 (2010)
8. Grosseck, G., Holotescu, C.: Can We use Twitter for Educational Activities. In: 4th International Scientific Conference. eLearning and Software for Education, Bucharest, April 17-18 (2008)
9. Jain, D., Sinhal, A., Gupta, N., Narwariya, P., Saraswat, D., et al.: Hiding Sensitive Association Rules without Altering the Support of Sensitive Item(s). International Journal of Artificial & Applications (IJAIA) 3(2) (2012)
10. Jansen, B.J., Zhang, M., Sobel, K., Chowdury, A.: Micro-blogging as Online Word of Mouth Branding. In: CHI 2009, pp. 3859–3864 (2009)
11. Koetsier, J.: Twitter Reaches 500M users, 140M in the U.S. VB News, July 30 (2012), `http://venturebeat.com/2012/07/30/twitter-reaches-500-million-users-140-million-in-the-u-s/` (retrieved September 22, 2012)

12. Liu, D.-R., Shih, M.-J., Liau, C.-J., Lia, C.-H.: Mining the Change of Event Trends for Decision Support in Environmental Scanning. Science Direct, Expert Systems with Application 36, 972–984 (2009)
13. Moen, P.: Data Mining Methods: Some Properties of Association Rules. Lecture notes, University of Helsinki/Dept. of Computer Science (2005), http://www.cs.helsinki.fi/u/ronkaine/tilome/luentomateriaali/TiLoMe-140305.pdf (retrieved on August 04, 2012)
14. Murthy, D.: Twitter: Microphone for the Masses? Media, Culture & Society 33(5), 779–789 (2011), doi:10.1177/0163443711404744
15. Nobil, A.: Is Twitter a Useful Tool for Journalists? Journal for Media Practice 11(1), 145–155 (2010)
16. Pak, A., Paroubek, P.: Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC 2010), May 19-21. European Language Resources Association (ELRA), Valletta (2010)
17. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. Now Publishers Inc., Hanover (2008)
18. Silva, I.S., Gomide, J., Barbosa, G.A., Santos, W., Veloso, A., Meira Jr., W., Ferreira, R.: Observatório da Dengue: Surveillance Based on Twitter Sentiment Stream Analysis (2011)
19. Song, H., Kim, J., Kim, S.: Mining the Change of Customer Behaviour in an Internet Shopping Mall. Expert Systems with Applications 21, 157–168 (2001)
20. Java, A., Song, X., Finin, T., Tseng, B.: Why We Twitter: Understanding Microblogging Usage and Communities. In: 9th WEBKDD and 1st SNA-KDD Workshop 2007, San Jose, California, USA, August 12. Copyright 2007 ACM (2007), 1-59593-444-8
21. Wang, X., Wei, F., Liu, X., Zhou, M., Zhang, M.: Topic Sentiment Analysis in Twitter: A graph-based hashtag sentiment classification approach. In: Proceedings of 20th ACM International Conference on Information and Knowledge Management, CIKM, USA, pp. 1031–1040 (2011)
22. Weng, J., Lim, E.-P., He, Q., Wing-ki Leung, C.: What do People Want in Microblogs? Measuring Interestingness of Hashtags in Twitter. In: Proceedings of the 10th IEEE International Conference on Data Mining (ICDM), Sydney, pp. 1121–1126

Identifying Uncertain Galaxy Morphologies Using Unsupervised Learning

Kieran Jay Edwards and Mohamed Medhat Gaber

School of Computing, University of Portsmouth
Hampshire, England, PO1 3HE, UK
kieran.edwards@myport.ac.uk, mohamed.gaber@port.ac.uk

Abstract. With the onset of massive cosmological data collection through mediums such as the Sloan Digital Sky Survey (SDSS), galaxy classification has been accomplished for the most part with the help of citizen science communities like Galaxy Zoo. However, an analysis of one of the Galaxy Zoo morphological classification data sets has shown that a significant majority of all classified galaxies are, in fact, labelled as "Uncertain". This has driven us to conduct experiments with data obtained from the SDSS database using each galaxy's right ascension and declination values, together with the Galaxy Zoo morphology class label, and the k-means clustering algorithm. This paper identifies the best attributes for clustering using a heuristic approach and, accordingly, applies an unsupervised learning technique in order to improve the classification of galaxies labelled as "Uncertain" and increase the overall accuracies of such data clustering processes. Through this heuristic approach, it is observed that the accuracy of classes-to-clusters evaluation, by selecting the best combination of attributes via information gain, is further improved by approximately 10-15%. An accuracy of 82.627% was also achieved after conducting various experiments on the galaxies labelled as "Uncertain" and replacing them back into the original data set. It is concluded that a vast majority of these galaxies are, in fact, of spiral morphology with a small subset potentially consisting of stars, elliptical galaxies or galaxies of other morphological variants.

Keywords: Astronomical Data Mining, K-means, Cluster Identification, Classification Accuracy, Galaxy Morphology.

1 Introduction

The fourth paradigm [1], to which it is now referred, describes the emergence of data mining within various scientific disciplines, including that of astronomy. The Sloan Digital Sky Survey [2] alone possesses, at present, over 1,000,000 galaxies, 30,000 stars and 100,000 quasars collated into several data sets. With such copious amounts of data being acquired from various astronomical surveys, it now becomes imperative that an automated model to processing this data be developed so as to be able to generate useful information. The goal of this approach is to then produce an outcome that will result in effective human

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 146–157, 2013.

learning. It is the process of characterizing the known, assigning the new and discovering the unknown in such a data-intensive discipline that encompasses what astronomical data mining is all about [3].

Various classification techniques such as Nave Bayes [4,5], C4.5 [6,7,8] and Artificial Neural Networks (ANN) [9] appear to be the more popular choices of methods when processing astronomical data. However, research carried out [10] involving calculating the Davies-Bouldin Validity Index (DBI) of the various attributes to determine the best combination for identifying correlations between morphological attributes and user-selected morphological classes motivated the direction of our research. A list of the top 10 attributes was presented and the best combinations of these, which produced the lowest DB Index values, were analyzed. It was ascertained that the larger the DBI value an attribute produced, the less useful it would be for clustering. Notably, these same attributes also proved less than useful in decision tree classification.

As an initial experiment, we obtained data for these 10 attributes from the Sloan Digital Sky Survey database for 2500 galaxies which were identified by their right ascension and declination through the Galaxy Zoo Classification data set. The k-means algorithm was then applied and evaluated using classes-to-clusters evaluation. However, despite using various subsets or combinations of these 10 attributes and re-clustering these sets reiteratively, the resulting accuracies never exceeded 55%. This encouraged us to then obtain 135 attributes from the SDSS database table from which the 10 originated, and apply a heuristic technique in order to find the best combination of those attributes with respect to their information gain levels.

In this paper, an investigation of how to select the best combination of attributes for clustering and determining the categories of these galaxies using an unsupervised approach is carried out. We also show that the heuristic technique applied to the attribute selection process, based on information gain levels, improves the classes-to-clusters accuracy by approximately 10-15%, thus further improving the classification of these galaxies.

The rest of the paper is organized as follows. Section 2 explores the various attempts at comparing algorithms and improving the classification process of astronomical data, while Section 3 details the clustering techniques, the focus mainly being on the k-means algorithm, which is used throughout this research. Section 4 provides a detailed discussion of the acquisition of the data sets; the pre-processing involved which includes the heuristic approach to attribute selection, and the unsupervised clustering experiments that were carried out. The results acquired from these experiments and the overall conclusion of this research together with a direction for future work is provided in Sections 5 and 6 respectively.

2 Related Work

A whole host of techniques and algorithms have already been applied to astronomical data sets with the goal of improving classification, including C4.5, Nave Bayes, Random Forest (RF) and Artificial Neural Networks (ANN).

ANNs are slowly being utilized more often as they have proven, with the correct training, to be an effective means of classification. One study [11], which took a supervised approach, trained an ANN to predict specific properties of galaxies, namely morphological classifications and redshifts, which proved reliable. Similarly, an ANN was trained [12] to identify broad absorption line quasars, a sub-class of active galactic nuclei. The results showed accuracies of approximately 92%. However, results of ANN implementations depend heavily on effective pre-processing. A comparison of different ANN algorithms [13] used to classify astronomical objects, also utilizing supervised learning, showed that the pre-processing methods used were insufficient and had an equally negative impact on both the algorithms that were being compared. Another issue with ANN algorithms is their inferiority when used with high dimensional data [15].

The Nave Bayes algorithm is probabilistic [14] in that it assumes that all attributes are statistically independent. As such, it works by assigning to each object it encounters the most probable target value. However, based on a classification comparison done [6] between the Nave Bayes algorithm, the RF algorithm and the C4.5 algorithm, it is observed that the Nave Bayes classification results, with its best accuracy of 43.62%, are nowhere near as robust as that of the RF algorithm's.

Classifying galaxies using the C4.5 algorithm to generate decision trees has also been the focus of much research. The main benefit of the C4.5 algorithm is the fact that it is efficient when dealing with numerical and nominal attributes alike. It has been applied [7] successfully in distinguishing between spiral and elliptical galaxies. It is noted that the global accuracy obtained from all C4.5 algorithm experiments consistently stayed above 96.2%. This involved using confidence levels of 0.1 as well as 0.25.

Random forests are also deemed suitable for dealing with astronomical data sets as they are designed for effective use with very large amounts of data. When used to classify stars, quasars and galaxies [15], the overall accuracy approximates between 91-95%, showing a significant improvement over the use of single tree classifiers such as the C4.5 algorithm. Classifying galactic images using RF [6] has also shown to outperform its C4.5 counterpart and the Nave Bayes algorithm.

3 Background

Clustering, in its unsupervised form, has always been one of the key areas in exploratory data analysis (e.g. astronomy). It is referred to [5], more commonly in astronomy circles, as "spatial clustering" or "angular clustering" on the sky. One of the main advantages of utilizing such a technique is the ability to discover hidden clusters or structures in the presented data sets. Clustering, in a general sense, is the process of partitioning a data set into groups based on similarity between attributes of objects. A clustering algorithm is considered consistent [16] if it outputs an effectively-defined partition. The issues that plague the various clustering techniques include the determination of the quality of these

partitions and the consistency of the overall results. The three subsections that follow will briefly describe a few of the various clustering techniques that have been developed, provide a more in-depth view of the k-means algorithm, and briefly describe cluster-to-class evaluation.

3.1 Clustering Techniques

EM (Expectation-Maximization) Clustering Algorithm - The EM algorithm is often used for clustering and is known especially for its ability to handle missing data. It is defined as an iterative means of calculating the maximum likelihood of parameters. With each cycle, there is an alternation between the expectation (E) and maximization (M) steps in which new parameter estimates, proven to not decrease the log-likelihood, are output. This process is reiterated until the best-fit maximum-likelihood solution over the initial model parameters is found [17].

Hierarchical Clustering Algorithm - The hierarchical clustering algorithm builds a hierarchy of clusters for analysis that can be achieved through either an agglomerative [18] or divisive [19] strategy (i.e. "bottom up" or "top down" approach respectively). Some of the choices of metrics, depending on the nature of the objective of clustering, can be the Euclidean distance, Manhattan distance, maximum distance or even cosine similarity. The advantage that this algorithm has lies in its ability to use any measure of distance so long as it is valid.

Spectral Clustering Algorithm - Spectral clustering techniques are used to solve problems in graph partitions where different measures require optimization. This involves a two-step process [20]: Taking the various data points from more "obvious" clusters and embedding them in a space, followed by the application of a classical clustering algorithm such as the k-means algorithm.

3.2 K-Means Algorithm

The k-means algorithm is one of the most popular clustering techniques available, used extensively in both industrial and scientific applications for cluster analysis. It is known [21] as a partitional or nonhierarchical clustering technique in which the aim is to partition n objects into k clusters where each object belongs to the cluster with the closest mean. This is an iterative, heuristic approach which starts with the assignment for each object as described in equation (1) given an initial set of k means $m_1^{(1)}, \ldots, m_k^{(1)}$, where each x_j is allocated to one $S_i^{(t)}$.

$$S_i^{(t)} = \{x_j : ||x_j - m_i^{(t)}|| \leq ||x_j - m_{i^*}^{(t)}|| \forall i^* = 1, \ldots, k\} \tag{1}$$

This is followed by the calculation of the new means which is to become the newly appointed centroid of the cluster as shown in equation (2).

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \tag{2}$$

Table 1. Galaxy Zoo Table 2 Data Set: Final Morphological Classifications

Category	No. of Galaxies
Uncertain	41556
Spiral	17747
Elliptical	6232

The iteration of these two steps will continue until convergence is achieved. When this occurs, the assignments of the centroids no longer change. The number of iterations required to achieve convergence can vary greatly which makes this algorithm potentially computationally intensive particularly with very large data sets. However, there are a number of variants of the k-means algorithm which address this problem [22,23], improving its efficiency.

3.3 Classes-to-Clusters Evaluation

In the various experiments carried out, we take an unsupervised approach by using the k-means algorithm together with classes-to-clusters evaluation in order to evaluate the resulting clustering of the data and determine its accuracy. In classes-to-clusters evaluation, the class label (i.e. Spiral, Elliptical and Uncertain) is first ignored and the clusters generated using only numerical data. The clusters are then assigned classes based on the largest number of objects in each cluster that fall into a certain class. A classification error and confusion matrix are then computed which shows the resulting accuracy of the process.

4 Methodology

We started by obtaining the galaxy morphological classification voting data from the Galaxy Zoo Table 2 [24] data set and observed that a significant majority of all galaxies, approximately 63%, have been classified as "Uncertain". Table 1 shows the final classification result.

This led us to obtain data for the 10 attributes that were calculated to have the lowest DBI index values [10] from the SDSS database [25]. The three flag attributes used in the Galaxy Zoo Table 2 data set to indicate the morphology of the galaxies were combined, in order to decrease sparseness, into one attribute labeled "CLASS" and then included together with the 10 attributes. Table 2 shows the list of the 10 attributes used for this initial experiment.

The relational algebraic query used to retrieve data from the SDSS database to obtain the data for the 10 attributes can be expressed as follows.

result = π σ $isoAGrad\_u*z/a.isoAGrad\_u*z$, σ $petroRad_u*z/a.petroRad\_u*z$, σa.texture_u, σ $isoA\_z*z/a.isoA\_z*z$, σ $lnLExp_u/a.lnLExp\_u$, σ $lnLExp\_g/a.lnLExp\_g$, σ $isoA\_u*z/a.isoA\_u*z$, σ $isoB\_z*z/a.isoB\_z*z$, σ $isoBGrad\_u*z/a.isoBGrad\_u*z$, σ $isoAGrad\_z*z/a.isoAGrad\_z*z$ ($u.up\_id = x.up\_id \wedge$ x.objID=p.objID $\wedge$ p.objID=a.objID ($x(\#x)$ | $\times$ | $u(\#upload)$ | $\times$ | p(PhotoTag) | $\times$ | a(PhotoObjAll)))

Table 2. The 10 Attributes with the Lowest DBI Index Values

Attribute	Description
*isoAGrad u*z*	Gradient of the isophotal major axis
*petroRad u*z*	Petrosian radius
texture u	Measurement of surface texture
*isoA z*z*	Isophotal major axis
lnLExp u	Log-likelihood of exponential profile fit (typical for a spiral galaxy)
lnLExp g	Log-likelihood of exponential profile fit (typical for a spiral galaxy)
*isoA u*z*	Isophotal major axis
*isoB z*z*	Isophotal minor axis
*isoBGrad u*z*	Gradient of the isophotal minor axis
*isoAGrad z*z*	Gradient of the isophotal major axis

Table 3. The Best Resulting Subset of the Original 10 Attributes

Attribute
*isoA z*z*
lnLExp g
*isoAGrad u*z*
*isoB z*z*

Accurate application of the morphological class labels to each of the galaxies in the data set before clustering was achieved by reference to each galaxy's right ascension and declination values. These produced, in the SDSS database query, the object ID for each galaxy which was then matched up to the object ID in the Galaxy Zoo Table 2 data set to obtain the correct label (i.e. Spiral, Elliptical or Uncertain). The k-means clustering algorithm was applied to the full data set of 3000 galaxies using classes-to-clusters evaluation with the value of k set to 3. This process was then repeated reiteratively using various subsets of the 10 attributes. The best resulting subset is shown in table 3.

The following R code used to apply the k-means algorithm to the data set and compare the clusters to classes.

```
Library(RWeka)
sdssTable <- read.csv(file=sdss.csv)
sdssTable2 <- sdssTable
results <- SimpleKMeans(sdssTable2[,-5], Weka_control(N=3))
results
table(predict(results), sdssTable\$CLASS)
```

It was originally thought that the reason for the low accuracies was due to the majority of the galaxies having been labeled as "Uncertain". An alternative clustering attempt, where 1000 of the 1763 galaxies labelled as "Uncertain" were removed, was carried out but proved ineffective as it showed no improvement whatsoever. In fact, the accuracy level dropped even further.

Table 4. The Best Combination of Attributes with Respective Information Gain Levels

Attribute	Information Gain	Attribute	Information Gain
expRad_g	0.2207	*isoAGrad_r*	0.0775
expRad_r	0.1965	*lnLDeV_z*	0.0716
expRad_i	0.1831	*texture_g*	0.0706
lnLDeV_g	0.1367	*isoPhiGrad_g*	0.0639
lnLDeV_r	0.1275	*texture_r*	0.0522
isoB_i	0.1206	*lnLDeV_u*	0.0428
isoB_r	0.1154	*texture_i*	0.0367
lnLExp_r	0.1002	*isoPhiGrad_i*	0.03
lnLExp_i	0.0986	*texture_u*	0.0153
isoBGrad_g	0.092	*isoColcGrad_r*	0.0115
petroRad_u	0.0834		
lnLExp_z	0.0822		

The objective of this paper is to be able to provide astronomers with a tool to effectively assign each galaxy to the right category as accurately as possible. With that in mind, we decided to re-query the SDSS database, this time obtaining 135 attributes, and apply a heuristic technique in order to obtain the best combination. This was achieved through the use of each of the attribute's information gain levels. Refer to the appendix for the full relational algebraic query that was used for data retrieval from the SDSS database.

4.1 Best Attribute Combination through Information Gain

After acquiring the 135 attributes for 5000 galaxies and pre-processing the data set, which involved removing selected attributes and objects that contained significant numbers of entries with the value -9999, the information gain level for all attributes was calculated and then listed in descending order. The heuristic technique was then employed. This involved clustering the data with the single attribute that possessed the highest information gain level together with the class label, using clusters-to-classes evaluation with the value of k set to 3. Once this 1st iteration completed, the attribute with the 2nd highest information gain level was then added in and the data re-clustered. If the accuracy level decreased, that 2nd attribute would then be removed and then the 3rd added. If the accuracy level remained the same or increased, that attribute would remain and the next attribute added on.

The final data set contained 4979 galaxies and 23 attributes. Table 4 lists the attributes and their information gain levels.

4.2 Hidden Cluster Discovery and Labeling by Unsupervised Clustering

After acquiring the best combination of attributes, various clustering experiments were carried out in an attempt to accurately classify the galaxies labelled

Table 5. The Iterative Clustering Results of the 10 attributes

No. of Attributes	Accuracy (%)	Within Cluster Sum of Squared Errors
1	50.8	0.31097208092561296
2	54.2	73.22356287236981
3	54.2	3.213059611539209
4	54.2	56.57163126388849
4	49.5333	3.213059611539209
5	49.4	5.1038948660063035
10	45.8	186.893316665896

as "Uncertain". In order to further analyze these galaxies, we split them into two clusters: "cluster0" and "cluster1". These were saved and placed back into the original data set. The "cluster0" and "cluster1" clusters were then re-labeled; starting with "cluster0" being re-labeled as "spiral" and "cluster1" as "elliptical", and then the data set was clustered with the value of k set to 2. The labels of "cluster0" and "cluster1" were then reversed and the process repeated.

5 Experimental Results

The results of the initial experiments done on the 10 attributes with the lowest *DBI Index* Values [10] with 3000 galaxies and the value of k set to 3 are shown in Table 5.

After the 4th attempt, the accuracy lowered whenever additional attributes were added so the best subset contained only 4 attributes in the end.

After utilizing our heuristic technique to obtain the best selection from the 135 attributes obtained from the SDSS database, we conducted several clustering experiments on our final data set consisting of 4979 galaxies and 23 attributes which involved separating clusters, re-clustering, re-labeling and re-combining them together. Table 6 shows the various k-means clustering results accordingly.

It is notable that the highest clustering accuracy of 82.627% was obtained when galaxies from both "cluster0" and "cluster1" were re-labelled as "Spiral" galaxies. Out of the 4979 galaxies in the complete data set, only 865 were incorrectly classified. Motivated by this boost in accuracy, we conducted another set of experiments using state-of-the-art classification techniques, namely Random Forest (RF) [26] and Support Vector Machines (SVM) [27]. Table 7 lists the results of these additional experiments.

The accuracies for all three algorithms, when all the galaxies from "cluster0" and "cluster1" are re-labelled as "Spiral", consistently outperform the rest of the experiments. With the number of trees set to 100, Random Forest provided an exceptional accuracy of 91.3838% which indicates two concluding remarks that we can state with certainty:

– A significant majority of the galaxies labelled as "Uncertain" are indisputably of spiral morphology.

Table 6. The Results of the various k-means Clustering Experiments

Data Set Type	Number of Galaxies Per Cluster			Accuracy (%)
	Spiral	Elliptical	Uncertain	
Full Data Set	1476	520	2983	65.6156
Spiral/Elliptical Only	1476	520	-	72.495
Uncertain Only	-	-	2983	78.9474
Cluster0 - Spiral / Cluster1 - Elliptical	2104	2875	-	63.0649
Cluster0 - Elliptical / Cluster1 - Spiral	3831	1148	-	77.2444
Cluster0 - Spiral / Cluster1 - Spiral	4459	520	-	82.627
Cluster0 - Elliptical / Cluster1 - Elliptical	1476	3503	-	68.4475

Table 7. The Additional Experiments Involving RF and SVM Classification Techniques

Data Set Type	Algorithm Accuracy (%)		
	k-means	RF	SVM
Cluster0 - Spiral / Cluster1 - Elliptical	63.0649	90.6005	86.9452
Cluster0 - Elliptical / Cluster1 - Spiral	77.2444	83.6513	77.9675
Cluster0 - Spiral / Cluster1 - Spiral	82.627	91.3838	89.6566
Cluster0 - Elliptical / Cluster1 - Elliptical	68.4475	83.089	78.3892

– There is another small subset of galaxies amongst those that are "Uncertain" that are either of elliptical morphology, are stars or possess an entirely different morphology type.

6 Conclusion and Future Work

Motivated by the fact that 60% of all galaxies in the Galaxy Zoo Table 2 data set are classified as "Uncertain", we attempted to introduce a means for astronomers to more efficiently and accurately classify these galaxies. We introduced a novel approach to accomplishing such a task by first utilizing a heuristic technique in order to obtain the best combination of attributes through their calculated information gain which serves to increase the clustering accuracy. We then conducted a series of experiments involving the clustering of the galaxies labelled as "Uncertain", saving their cluster assignments and then re-introducing them back into the original data set. We have shown that the highest accuracy (82.627%) was

obtained when all the galaxies from "cluster0" and "cluster1" were re-labelled as "Spiral" galaxies. Applying the Random Forest and SVM classification algorithms over all the original experiments further reinforced this finding. There is no doubt that a majority of the galaxies labelled as "Uncertain" in our data set are, in fact, of spiral morphology.

Avenues for future work include a large-scale processing of the right ascension and declination values for all 65535 galaxies and re-running the experiments that were performed in this paper.

References

1. Ball, N.M., Brunner, R.J.: Data Mining and Machine Learning in Astronomy. International Journal of Modern Physics D, 61 (2009)
2. Stoughton, C., Lupton, R.H., Bernardi, M., Blanton, M.R., Burles, S., Castander, F.J., et al.: Sloan Digital Sky Survey: Early Data Release. The Astronomical Journal 123(1), 485 (2007)
3. Borne, K.: Scientific Data Mining in Astronomy. In: Next Generation of Data Mining, pp. 91–114 (2009)
4. Henrion, M., Mortlock, D.J., Hand, D.J., Gandy, A.: A Bayesian Approach to Star-Galaxy Classification. In: Monthly Notices of the Royal Astronomical Society, pp. 2286–2302 (2011)
5. Kamar, E., Hacker, S., Horvitz, E.: Combining Human and Machine Intelligence in Large-Scale Crowdsourcing. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, pp. 467–474 (2012)
6. de la Calleja, J., Fuentes, O.: Automated Classification of Galaxy Images. In: Negoita, M.G., Howlett, R.J., Jain, L.C. (eds.) KES 2004. LNCS (LNAI), vol. 3215, pp. 411–418. Springer, Heidelberg (2004)
7. Gauci, A., Adami, K.Z., Abela, J.: Machine Learning for Galaxy Morphology Classification. arXiv:1005.0390, pp. 1–9 (2010)
8. Vasconcellos, E.C., de Carvalho, R.R., Gal, R.R., LaBarbera, F.L., Capelato, H.V., Velho, H.F.C., Ruiz, R.S.R.: Decision Tree Classifiers for Star/Galaxy Separation. The Astronomical Journal 141, 189 (2011)
9. Banerji, M., Lahav, O., Lintott, C.J., Abdalla, F.B., Schawinski, K., Bamford, S.P., Andreescu, D., Murray, P., Raddick, M.J., Slosar, A., Szalay, A., Thomas, D., Vandenberg, J.: Galaxy Zoo: Reproducing Galaxy Morphologies Via Machine Learning. In: Monthly Notices of the Royal Astronomical Society, pp. 342–353 (2010)
10. Baehr, S., Vedachalam, A., Borne, K.D., Sponseller, D.: Data Mining the Galaxy Zoo Mergers. In: 2010 Conference on Intelligent Data Understanding (2010)
11. Ball, N.M., Loveday, J., Fukugita, M., Nakamura, O., Okamura, S., Brinkmann, J., Brunner, R.J.: Galaxy Types in the Sloan Digital Sky Survey Using Supervised Artificial Neural Networks. In: Monthly Notices of the Royal Astronomical Society, pp. 1038–1046 (2004)
12. Scaringi, S., Cottis, C.E., Knigge, C., Goad, M.R.: Broad Absorption Line Quasar Catalogues with Supervised Neural Networks. arXiv:0810.4396 (2008)
13. Bazell, D., Peng, Y.: A Comparison of Neural Network Algorithms and Preprocessing Methods for Star-Galaxy Discrimination. The Astrophysical Journal Supplement Series 116(1), 47 (2009)

14. Frank, E., Hall, M., Pfahringer, B.: Locally Weighted Nave Bayes. In: Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence, pp. 249–256 (2002)
15. Gao, D., Zhang, Y.X., Zhao, Y.H.: Random Forest Algorithm for Classification of Multi-wavelength Data. Research in Astronomy and Astrophysics 9(2), 220 (2009)
16. Von Luxburg, U., Bousquet, O., Belkin, M.: Limits of Spectral Clustering. In: Advances in Neural Information Processing Systems (NIPS), pp. 857–864 (2005)
17. Bradley, P.S., Fayyad, U., Reina, C.: Scaling EM (Expectation-Maximization) Clustering to Large Databases. In: Microsoft Research (1998)
18. Karypis, G., Han, E.H., Kumar, V.: Chameleon: Hierarchical Clustering Using Dynamic Modeling. Computer 32(8), 68–75 (1999)
19. Ding, C., He, X.: Cluster Merging and Splitting in Hierarchical Clustering Algorithms. In: Proceedings of the 2002 IEEE International Conference on Data Mining, pp. 139–146 (2002)
20. Bengio, Y., Paiement, J.F., Vincent, P., Delalleau, O., Le Roux, N., Ouiment, M.: Out-of-Sample Extensions for Lle, Isomap, Mds, Eigenmaps and Spectral Clustering. In: Advances in Neural Information Processing Systems, vol. 16, pp. 177–184 (2004)
21. Huang, Z.: Extensions to the K-means Algorithm for Clustering Large Data Sets with Categorical Values. Data Mining and Knowledge Discovery 2(3), 283–304 (1998)
22. Alsabti, K., Ranka, S., Singh, V.: An Efficient K-Means Clustering Algorithm. Electrical Engineering and Computer Science (43) (1997)
23. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An Efficient K-Means Clustering Algorithm: Analysis and Implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 881–892 (2002)
24. Lintott, C., Schawinski, K., Bamford, S., Slosar, A., Land, K., Thomas, D., et al.: Galaxy Zoo 1: Data Release of Morphological Classifications for Nearly 900.000 Galaxies. Monthly Notices of the Royal Astronomical Society 410(1), 166–178 (2011)
25. Abazajian, K.N., Adelman-McCarthy, J.K., Agueros, M.A., Allam, S.S., Prieto, C.A., An, D., et al.: The Seventh Data Release of the Sloan Digital Sky Survey. The Astrophysical Journal Supplement Series, 543 (2009)
26. Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)
27. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning 20(3), 273–297 (1995)

Appendix: SDSS Database Query (135 Attributes)

The relational algebraic query for the data of the 135 attributes that was used for data retrieval from the SDSS database.

result = π a.petroMag_u, a.petroMag_g, a.petroMag_r, a.petroMag_i, a.petroMag_z,
a.petroRad_u, a.petroRad_g, a.petroRad_r, a.petroRad_i, a.petroRad_z, a.petroR90_u, a.petroR90_g, a.petroR90_r, a.petroR90_i, a.petroR90_z, a.isoRowc_u, a.isoRowc_g, a.isoRowc_r, a.isoRowc_i, a.isoRowc_z, a.isoRowcGrad_u, a.isoRowcGrad_g, a.isoRowcGrad_r, a.isoRowcGrad_i, a.isoRowcGrad_z, a.isoColc_u, a.isoColc_g, a.isoColc_r, a.isoColc_i, a.isoColc_z, a.isoColcGrad_u, a.isoColcGrad_g, a.isoColcGrad_r, a.isoColcGrad_i, a.isoColcGrad_z, a.isoA_u, a.isoA_g, a.isoA_r, a.isoA_i, a.isoA_z, a.isoB_u, a.isoB_g, a.isoB_r, a.isoB_i, a.isoB_z, a.isoAGrad_u, a.isoAGrad_g, a.isoAGrad_r, a.isoAGrad_i, a.isoAGrad_z, a.isoBGrad_u, a.isoBGrad_g, a.isoBGrad_r, a.isoBGrad_i, a.isoBGrad_z, a.isoPhi_u, a.isoPhi_g, a.isoPhi_r, a.isoPhi_i, a.isoPhi_z, a.isoPhiGrad_u, a.isoPhiGrad_g, a.isoPhiGrad_r, a.isoPhiGrad_i, a.isoPhiGrad_z, a.deVRad_u, a.deVRad_g, a.deVRad_r, a.deVRad_i, a.deVRad_z, a.deVAB_u, a.deVAB_g, a.deVAB_r, a.deVAB_i, a.deVAB_z, a.deVPhi_u, a.deVPhi_g, a.deVPhi_r, a.deVPhi_i, a.deVPhi_z, a.deVMag_u, a.deVMag_g, a.deVMag_r, a.deVMag_i, a.deVMag_z, a.expRad_u, a.expRad_g, a.expRad_r, a.expRad_i, a.expRad_z, a.expAB_u, a.expAB_g, a.expAB_r, a.expAB_i, a.expAB_z, a.expPhi_u, a.expPhi_g, a.expPhi_r, a.expPhi_i, a.expPhi_z, a.expMag_u, a.expMag_g, a.expMag_r, a.expMag_i, a.expMag_z, a.modelMag_u, a.modelMag_g, a.modelMag_r, a.modelMag_i, a.modelMag_z, a.texture_u, a.texture_g, a.texture_r, a.texture_i, a.texture_z, a.lnLExp_u, a.lnLExp_g, a.lnLExp_r, a.lnLExp_i, a.lnLExp_z, a.lnLDeV_u, a.lnLDeV_g, a.lnLDeV_r, a.lnLDeV_i, a.lnLDeV_z, a.fracDeV_u, a.fracDeV_g, a.fracDeV_r, a.fracDeV_i, a.fracDeV_z, a.dered_u, a.dered_g, a.dered_r, a.dered_i, a.dered_z(u.up_id=x.up_id $\wedge$ x.objID=p.objID $\wedge$ p.objID=a.objID (x(#x) |$\times$| u(#upload) | $\times$ | p(PhotoTag) | $\times$ | a(PhotoObjAll)))

Kernel Estimation of Regression Functions in the Boundary Regions

Tomasz Gałkowski

Institute of Computational Intelligence
Czestochowa University of Technology, Czestochowa, Poland
tomasz.galkowski@iisi.pcz.pl

Abstract. The article refers to the problem of regression functions estimation in the points near the edges of their domain. We investigate the model $y_i = R(x_i) + \epsilon_i$, $i = 1, 2, \ldots n$, where x_i is assumed to be the set of deterministic inputs, $x_i \in D$, y_i is the set of probabilistic outputs, and ϵ_i is a measurement noise with zero mean and bounded variance. $R(.)$ is a completely unknown function. The possible clue of finding unknown function is to apply the algorithms based on Parzen kernel [5], [12]. The commonly known inconvenience of these algorithms is that the error of estimation dramatically increases if the point of estimation x is coming up to the left or right bound of interval D.

The main result of this paper is a new, original algorithm (named NMS) based on integral version of Parzen methods for estimation of edge values of a function R. The cross-validation-like technique is used in this procedure. The results of numerical experiments are presented.

1 Introduction

In literature various nonparametric algorithms have been proposed for modelling and classification in stationary [1-2], [6], [16-18], [23-24], [27-29], [31-32], quasi-stationary [19], [21] and time-varying [7], [20], [22], [25-26] environments.

The article refers to the problem of regression functions estimation in the points near the edges of their domain. We investigate the model of type $y_i = R(x_i) + \epsilon_i$, $i = 1, 2, \ldots n$, where x_i is assumed to be the set of deterministic inputs, $x_i \in D$, y_i is the set of probabilistic outputs, and ϵ_i is a measurement noise with zero mean and bounded variance. $R(.)$ is a completely unknown function. We have no assumption neither on its shape (like e.g. in the spline methods) nor on any mathematical formula depending on certain set of parameters to be found (so-called parametric approach). This article consider an approach known from literature as a non-parametric estimation. The possible approaches of finding unknown function are based on Parzen kernel [3] [5] [14] or methods derived from orthogonal series [29]. Let us mention that the Parzen kernel methods are much more often applied and analysed for estimation of probability density functions and/or regressions with probabilistic input than for deterministic case.

Applications based on above bring satisfying results when the estimate is taken in the interior of the function $R(.)$ domain, i.e. the error of estimation

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 158–166, 2013.

dramatically increases if the point of estimation x is coming up to the left or right bound of interval D in which measurements of R were taken, depending on some smoothing parameter a_n.

What is the reason of that phenomenon? Simple though intuitive answer is that the estimator when works in the boundaries has less number of measurements (input data) than in the interior of D. For the left boundary for instance, the kernel is applying only points belonging to the interval $(0,\ a_n]$, however the interval $[-a_n,0)$ shows complete absence of them. In the algorithm dominate measurements in the points x_i for low values of index i (i.e. i=1,2,3,...) and the estimates are strongly biased of noises, which are the subjects of specific amplification there. This fact results in estimates of $R(.)$ walking away to noised measurement values (see Fig. 1.), when a_n tends to zero.

This fact has its consequence in known theorems on bias of the estimates. They are valid in the interior of D only. No results on any estimation method taken in the edge points exactly are known to the author of this paper.

There are a lot of efforts to solve the above problem in the boundary regions. The first are taken by Gasser et al. [5], followed by Müller [13] and Schuster [30]. In the last years we may observe that several authors still try to improve the previous results e.g. Karunamuni et al. [9] [10], Kyung-Joon et al. [11], Poměnková-Dluhá [15], Chen [1], Hazelton et al. [8], Zhang et al. [33] [34] [35].

The main result of this paper is an algorithm based on integral version of Parzen methods for an estimation of edge values of a function R. The cross-validation-like technique is used in this procedure. The numerical experiment results have been presented.

2 Idea of the Algorithm of Estimation of Edge Values of Function

Nonparametric algorithm of identification of unknown function $R(.)$ is in the form:

$$\hat{R}_n(x) = \frac{1}{a_n} \sum_{i=1}^{n} y_i \int_{D_i} K\left(\frac{x-u}{a_n}\right) du \tag{1}$$

where $K(.)$ is the kernel function described by (2), a_n is a smoothing parameter depending on the number of observations n. Interval D is partitioned into n disjunctive segments D_i such that $\cup D_i = [0,1], D_i \cap D_j = \emptyset$ for $i \neq j$. The measurement points x_i are chosen from D_i, i.e.: $x_i \in D_i$. Kernel function:

$$\left.\begin{array}{ll} \text{(i)} & K(t) = 0, \text{ for } t \notin (-\tau, \tau), \tau > 0, \\ \text{(ii)} & \int_{-\tau}^{\tau} K(t)\, \mathrm{d}t = 1 \\ \text{(iii)} & |K(t)| < \infty \end{array}\right\} \tag{2}$$

In the planning-phase of the experiment we choose the points x_i, for taking the measurements of R - in the presence of noise. We try to grant an uniform representation of function R in domain D. Thus, the standard assumption in theorems on convergence is that the $max\,|D_i|$ tends to zero if n tends to infinity

(see e.g. [2] [3] [5]). Furthermore, we may guess that in the set of pairs (x_i,y_i) there is - somehow encoded - the information on properties of function R, like its smoothness, for instance. Such supposition is underlying in such approaches like Markov chains - in the previous sequence of data there is the regularity helping in forecasting of future data.

Some papers describe methods using artificially expanded set of data e.g. by multinomial extension of function [33] [34], or by mirrored reflection of held data [30] [13], also by special kernel functions used in the boundary region [5] [35]. Mirrored reflection of data implies assumption that the estimated function has local extreme (minimum or maximum) in the edge point ($x = 0$ or $x = 1$). This means that the first derivative of the function $R(.)$ is equal to zero. Of course, this is the strong limitation of class of unknown function.

The main idea presented in this article is basing on auxiliary set of points outside domain D obtained by the original method of reflection of held data relatively to edge points. This reflection is "negative" and additionally "shifted" with the properly selected constant. The negative-mirror-shifted (NMS) algorithm is detailed in sequent section. Without loss of generality we assume construction of the expansion of function $R(.)$ in the left boundary; the algorithm can be used analogously in the opposite (right) end of the interval D.

3 The NMS Algorithm

The function $R(.)$ extended beyond the left edge $x = 0$ is defined in the expanded interval [-1,1] as follows:

$$\tilde{R}(x) = \begin{cases} R(x) & \text{for } x \in (0,1] \\ -R(-x) + 2S & \text{for } x \in [-1,0] \end{cases} \tag{3}$$

Let us mention that this method is similar to the odd expansion of function defined in finite interval in order to apply the Fourier series theorem.

The crucial problem is to determine the shift value S. Let us define the following loss function:

$$\mathcal{L}(S) = \int_{-1}^{1} \left[\hat{R}_n(x,S) - \tilde{R}(x,S)\right]^2 \mathrm{d}x \tag{4}$$

This function is a measure of distance between the expanded regression function $\tilde{R}(x)$ and its estimate $\hat{R}(x)$ taken in the expanded interval [-1,1], where

$$\hat{R}_n(x,S) = \frac{1}{a_n} \sum_{i=-n}^{+n} y_i \int_{D_i} K\left(\frac{x-u}{a_n}\right) \mathrm{d}u \tag{5}$$

For negative subscripts i we assign

$$y_{-i} = -y_i + 2S, \quad x_{-i} = -x_i \tag{6}$$

and, if $D_i = [d_{i-1}, d_i]$ then $D_{-i} = [-d_i, -d_{i-1}]$.

The problem is to minimize this loss function to find optimal S.

Unfortunately, we don't know the function $R(.)$ so, it is impossible to calculate the value of function (4), so far.

Moreover, because we can't use the true (exact) values of $R(.)$ let us to use the measurements y_i from experiment. Then the estimates $\hat{R}(x, S)$ should be taken in the points x_i. To assure the independence of the estimate and the observation y_i of $R(.)$ in the points x_i we shouldn't put into integral (4) the elements y_i when the estimate $\hat{R}(.)$ is taken in the corresponding points x_i ($\hat{R}(x_i)$).

Let us define the auxiliary "skip-one-out" estimator of function $\tilde{R}(.)$:

$$\hat{R}_{n,j}(x, S) = \frac{1}{a_n} \sum_{\substack{i=-n \\ i \neq j}}^{+n} y_i \int_{D_i} K\left(\frac{x-u}{a_n}\right) du \tag{7}$$

Let us to replace the integral in (4) by the sum; use the defined above skip-one-out estimator, and substitute the unknown values of $\tilde{R}(x_i, S)$ with the measurements y_i, finally obtaining:

$$\tilde{L}(S) = \sum_{j=1}^{n'} \left[\hat{R}_{n,j}(x_j, S) - y_j\right]^2 \tag{8}$$

where we assume that $n' = n$.

Now the independence of estimator and observation is fulfilled. The problem of finding S is now reduced to the problem of minimizing of the expression (8) with respect to S. Such technique of using measurement data instead of true - but unknown - values of function is known as a cross-validation method. The expression (8) could be finally rewritten as:

$$\tilde{L}(S) = \sum_{j=1}^{n'} [P_{1j} + 2 \cdot S \cdot P_{2j} - y_j]^2 \tag{9}$$

where

$$P_{1j} = \frac{1}{a_n} \sum_{\substack{i=-n \\ i \neq j}}^{+n} \operatorname{sgn}(i) \cdot y_{|i|} \int_{D_i} K\left(\frac{x_j - u}{a_n}\right) du \tag{10}$$

and

$$P_{2j} = \frac{1}{a_n} \sum_{i=-n}^{-1} \int_{D_i} K\left(\frac{x_j - u}{a_n}\right) du \tag{11}$$

By differentiating expression (9) relating to S, and by fulfilling condition $\tilde{L}' = 0$ one may obtain the estimate S^*:

$$S^* = \frac{\sum_{j=1}^{n'} (y_j - P_{1j}) \cdot P_{2j}}{2 \sum_{j=1}^{n'} P_{2j}^2} \tag{12}$$

We now adopt the estimated value S^* in the negatively mirrored expanded set of measurements:

$$\begin{gathered} \left[(x_{-n}, (y_{-n} + 2S^*)), \left(x_{-(n-1)}, (y_{-(n-1)} + 2S^*)\right), ... \right. \\ ..., (x_{-1}, (y_{-1} + 2S^*)), (0, S^*), (x_1, y_1), ... \\ \left. ..., (x_{n-1}, y_{n-1}), (x_n, y_n) \right] \end{gathered} \tag{13}$$

New algorithm of estimation of the regression function, working with the expanded data set described by (13), is defined as follows:

$$\hat{R}_n(x) = \frac{1}{a_n} \sum_{i=-n}^{+n} y_i \int_{D_i} K\left(\frac{x-u}{a_n}\right) du \tag{14}$$

It works in the points arbitrarily close to the left edge of interval D.

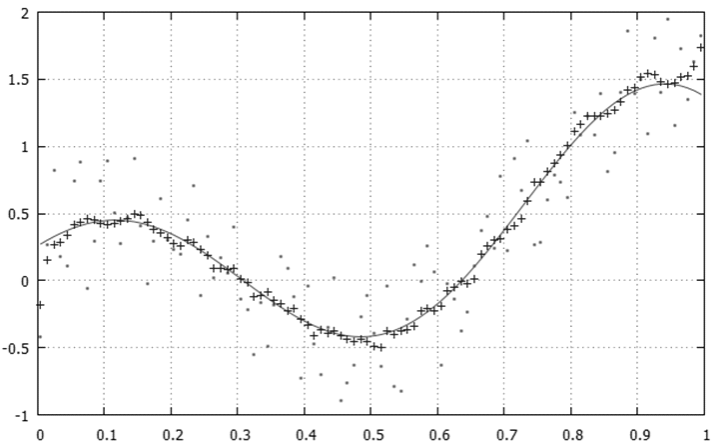


Fig. 1. Regression function $R(x)$ and its estimates

4 Simulation Study

The results of the simulation experiment are shown in the Figures 1-4. Figure 1. shows the example regression function $R(x)$ (continuous line) estimated in interval $D = [0, 1]$ (points marked with $+$) basing on measurement data set - marked with dots.

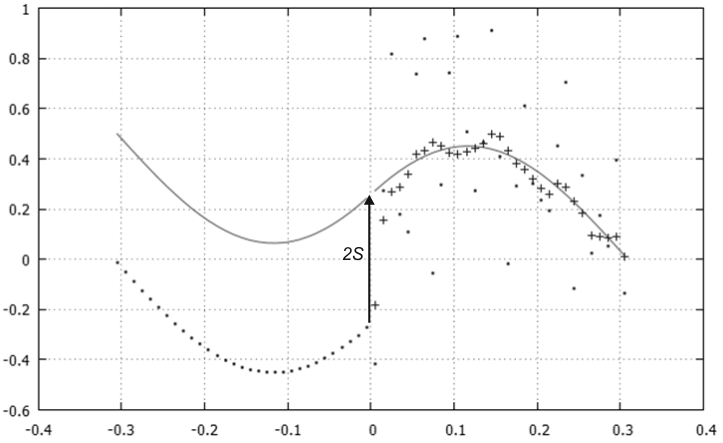


Fig. 2. Illustration of the negative-mirror-shifted (NMS) algorithm

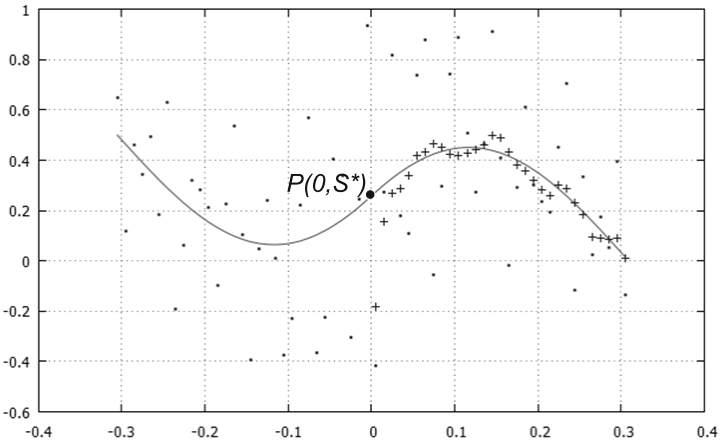


Fig. 3. Expanded set of input data

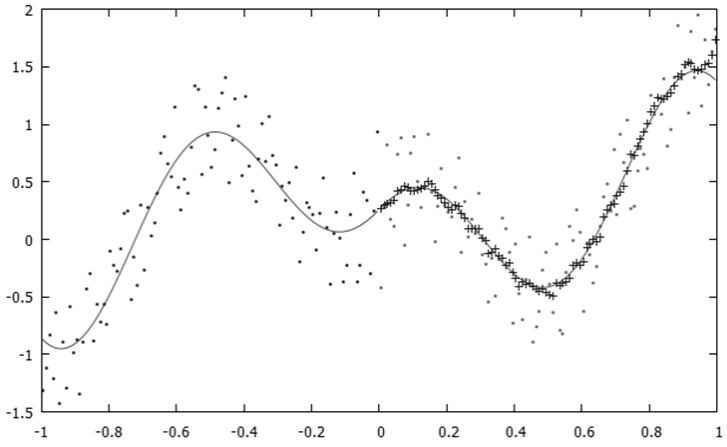


Fig. 4. Estimation of function $R(x)$ using NMS algorithm

Figure 2. illustrates the negative mirror method shifted with constant S. In the Figure 3. we can see resulting diagram with optimal value of S^* determined by equation (12).

Figure 4. shows the resulting estimation when modified (expanded) set of input data is used i.e. negatively mirrored and shifted with the optimal constant S^*.

5 Remarks and Extensions

The new NMS method based on the Parzen kernel algorithm for estimation of regression function in deterministic case in boundary regions has been proposed. By using our procedure it is possible particularly to determine the estimate of value of unknown function exactly in the edge point of domain. The graphical results of simulation let us to observe that the boundary effect is now strongly reduced or eliminated. For some analysis on convergence and bias of such estimator see [4]. One of the possible future extension of this work is to construct method for local prediction of function algorithm in regions near outside the domain.

References

1. Chen, S.X.: Beta kernel estimators for density functions. Journal of Statistical Planning and Inference 139, 2269–2283 (2009)
2. Gałkowski, T., Rutkowski, L.: Nonparametric recovery of multivariate functions with applications to system identification. Proceedings of the IEEE 73, 942–943 (1985)

3. Gałkowski, T., Rutkowski, L.: Nonparametric fitting of multivariable functions. IEEE Transactions on Automatic Control AC-31, 785–787 (1986)
4. Gałkowski, T.: Nonparametric estimation of boundary values of functions. Archives of Control Science 3(1-2), 85–93 (1994)
5. Gasser, T., Muller, H.G.: Kernel estimation of regression functions. Lecture Notes in Mathematics, vol. 757, pp. 23–68. Springer, Heidelberg (1979)
6. Greblicki, W., Rutkowski, L.: Density-free Bayes risk consistency of nonparametric pattern recognition procedures. Proceedings of the IEEE 69(4), 482–483 (1981)
7. Greblicki, W., Rutkowska, D., Rutkowski, L.: An orthogonal series estimate of time-varying regression. Annals of the Institute of Statistical Mathematics 35(1), 215–228 (1983)
8. Hazelton, M.L., Marshall, J.C.: Linear boundary kernels for bivariate density estimation. Statistics and Probability Letters 79, 999–1003 (2009)
9. Karunamuni, R.J., Alberts, T.: On boundary correction in kernel density estimation. Statistical Methodology 2, 191–212 (2005)
10. Karunamuni, R.J., Alberts, T.: A locally adaptive transformation method of boundary correction in kernel density estimation. Journal of Statistical Planning and Inference 136, 2936–2960 (2006)
11. Kyung-Joon, C., Schucany, W.R.: Nonparametric kernel regression estimation near endpoints. Journal of Statistical Planning and Inference 66, 289–304 (1998)
12. Marshall, J.C., Hazelton, M.L.: Boundary kernels for adaptive density estimators on regions with irregular boundaries. Journal of Multivariate Analysis 101, 949–963 (2010)
13. Müller, H.G.: Smooth optimum kernel estimators near endpoints. Biometrika 78, 521–530 (1991)
14. Parzen, E.: On estimation of a probability density function and mode. Analysis of Mathematical Statistics 33(3), 1065–1076 (1962)
15. Poměnková-Dluhá, J.: Edge Effects of Gasser-Müller Estimator, Mathematica 15, Brno, Masaryk University, pp. 307-314 (2004)
16. Rafajłowicz, E.: Nonparametric least squares estimation of a regression function Statistics. A Journal of Theoretical and Applied Statistics 19(3), 349–358 (1988)
17. Rafajłowicz, E., Schwabe, R.: Halton and Hammersley sequences in multivariate nonparametric regression. Statistics and Probability Letters 76(8), 803–812 (2006)
18. Rutkowski, L.: Sequential estimates of probability densities by orthogonal series and their application in pattern classification. IEEE Transactions on Systems, Man, and Cybernetics SMC-10(12), 918–920 (1980)
19. Rutkowski, L.: On Bayes risk consistent pattern recognition procedures in a quasi-stationary environment. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-4(1), 84–87 (1982)
20. Rutkowski, L.: On nonparametric identification with prediction of time-varying systems. IEEE Transactions on Automatic Control AC-29, 58–60 (1984)
21. Rutkowski, L.: Nonparametric identification of quasi-stationary systems. Systems and Control Letters 6, 33–35 (1985)
22. Rutkowski, L.: Real-time identification of time-varying systems by non-parametric algorithms based on Parzen kernels. International Journal of Systems Science 16, 1123–1130 (1985)
23. Rutkowski, L.: A general approach for nonparametric fitting of functions and their derivatives with applications to linear circuits identification. IEEE Transactions Circuits Systems CAS-33, 812–818 (1986)
24. Rutkowski, L.: Sequential pattern recognition procedures derived from multiple Fourier series. Pattern Recognition Letters 8, 213–216 (1988)

25. Rutkowski, L.: Application of multiple Fourier series to identification of multivariable nonstationary systems. International Journal of Systems Science 20(10), 1993–2002 (1989)
26. Rutkowski, L.: Non-parametric learning algorithms in the time-varying environments. Signal Processing 18(2), 129–137 (1989)
27. Rutkowski, L., Rafajłowicz, E.: On global rate of convergence of some nonparametric identification procedures. IEEE Transaction on Automatic Control AC-34(10), 1089–1091 (1989)
28. Rutkowski, L.: Identification of MISO nonlinear regressions in the presence of a wide class of disturbances. IEEE Transactions on Information Theory IT-37, 214–216 (1991)
29. Rutkowski, L.: Multiple Fourier series procedures for extraction of nonlinear regressions from noisy data. IEEE Transactions on Signal Processing 41(10), 3062–3065 (1993)
30. Schuster, E.F.: Incorporating Support Constraints Into Nonparametric Estimators of Densities. Communications in Statistics, Part A - Theory and Methods 14, 1123–1136 (1985)
31. Skubalska-Rafajłowicz, E.: Pattern recognition algorithms based on space-filling curves and orthogonal expansions. IEEE Transactions on Information Theory 47(5), 1915–1927 (2001)
32. Skubalska-Rafajłowicz, E.: Random projection RBF nets for multidimensional density estimation. International Journal of Applied Mathematics and Computer Science 18(4), 455–464 (2008)
33. Zhang, S., Karunamuni, R.J.: On kernel density estimation near endpoints. Journal of Statistical Planning and Inference 70, 301–316 (1998)
34. Zhang, S., Karunamuni, R.J.: Deconvolution boundary kernel method in nonparametric density estimation. Journal of Statistical Planning and Inference 139, 2269–2283 (2009)
35. Zhang, S., Karunamuni, R.J.: Boundary performance of the beta kernel estimators. Nonparametric Statistics 22, 81–104 (2010)

Instance Selection in Logical Rule Extraction for Regression Problems

Mirosław Kordos[1], Szymon Białka[1], and Marcin Blachnik[2]

[1] University of Bielsko-Biala, Department of Mathematics and Computer Science, Bielsko-Biała, Willowa 2, Poland
mkordos@ath.bielsko.pl

[2] Silesian University of Technology, Department of Management and Informatics, Katowice, Krasinskiego 8, Poland
marcin.blachnik@polsl.pl

Abstract. The paper presents three algorithms of instance selection for regression problems, which extend the capabilities of the CNN, ENN and CA algorithms used for classification tasks. Various combinations of the algorithms are experimentally evaluated as data preprocessing for regression tree induction. The influence of the instance selection algorithms and their parameters on the accuracy and rules produced by regression trees is evaluated and compared to the results obtained with tree pruning.

1 Introduction

1.1 Instance Selection in Classification Problems

The reasons for reducing the number of instance in the training set include: noise reduction by elimination outliers, reducing the data set size and improving generalization by eliminating instances that are too similar to each other, faster training of the model an a smaller dataset and faster prediction in of the model, especially in the case of lazy-learning algorithms, as k-NN. The early research in that area of instance selection in classification tasks lead to the Condensed Nearest Neighbor rule (CNN) [1] and Edited Nearest Neighbor rule (ENN) [2]. Another algorithm called CA, which can be thought of as an improved version of CNN was proposed by Chang [3]. In the following years, other, more complex algorithms were developed, such as Drop1-5 [4], IB3, Gabriel Editing (GE) and Relative Neighborhood Graph Editing (RNGE), Iterative Case Filtering (ICF), ENRBF2, ELH, ELGrow and Explore [5]. A large survey including almost 70 different algorithms of instance selection for classification tasks can be found in [6]. An interesting idea was proposed by Jankowski and Grochowski in [7]; to use the algorithms as instance filters for other machine learning algorithms like SVM, decision trees etc. By filtering noisy and compacting redundant examples they were able to improve the quality and speed of other classification algorithms.

1.2 Challenges in Regression Tasks

The instance selection issue for regression tasks is much more complex. In classification tasks only the boundaries between classes must be determined, while in regression

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 167–175, 2013.

tasks the output value must be assessed at each point of the input space. Moreover, in classification tasks there are at most several different classes, while in regression tasks, the output of the system is continuous, so there is an unlimited number of possible values to be predicted by the system. That is the reason why the dataset compression obtained by instance selection can be much higher in classification than in non-linear regression problems. The decision about rejection of a given vector in classification tasks can be made based on a right or wrong classification of the vector. In regression problems, rather a threshold defining the difference between the predicted and the actual value should be set. Determining the threshold (which is rather a function than a constant value) is an issue specific only to regression tasks. Another issue is the error measure, which in classification tasks is very straightforward, while in regression tasks, it can be defined in several ways and in practical solutions not always the simple error definitions as the MSE (mean square error) work best [8]. Because of the challenges, there were very few approaches in the literature to instance selection for regression problems. Moreover, the approaches were verified only on artificial datasets generated especially for the purpose of testing the algorithms. Zhang [9] presented a method to select the input vectors while calculating the output with k-NN. Tolvi [10] presented a genetic algorithm to perform feature and instance selection for linear regression models. In their works Guillen et al. [11] discussed the concept of mutual information used for selection of prototypes in regression problems.

1.3 Decision Trees

Clear logical rules are crucial factors in some practical implementations, where the decision must not only be made by the system but it also must be explained to humans [8,12]. The advantage of regression trees is their ability to easily generate comprehensive logical rules from data in a way easily to understand. For that reason in our practical implementations of computational intelligence to technological process optimization we always used a decision tree, usually as one of the components of a hybrid model. To keep the logical rules simple we use a univariate tree in all non-terminal nodes. Only the value assigned to the terminal leaves can be either a mean value of the vectors in the leaf or a linear regression, which includes no more than three most significant features in the leaf. Instance selection influence on classification trees were studied in [13] and [14]. In the current work we study that in case of regression trees.

2 Instance Selection Algorithms

2.1 ENN, CNN and CA Instance Selection Algorithms for Classification Problems

The CNN (Condensed Nearest Neighbor) algorithm was proposed by Hart [1]. The purpose of CNN is to reject these instances, which do not bring any additional information into the classification process. The algorithm starts with only one randomly chosen instance from the original dataset T. And this instance is added to the new dataset P. Then each remaining instance from T is classified with the k-NN algorithm, using the

k nearest neighbors from the dataset P. Only if the classification is wrong - the instance is added to P. The ENN (Edited Nearest Neighbor) algorithm was created by Wilson [2]. ENN works as a noise filter. The main idea of the algorithm is to remove a given instance if its class is different than the majority class of its neighbors. ENN starts with the whole original training set T. Each instance, which is wrongly classified by its k nearest neighbors is removed from the dataset, as it is supposed to be an outlier. In repeated ENN, the process of ENN is iteratively repeated as long as there are any instances wrongly classified. In all k-NN algorithm, the ENN is repeated for all k from $k = 1$ to k_max. The CA algorithm [3] works in a similar way to CNN. However, instead of rejecting one example, it replaces two closest examples of the same class by a single example situated in the middle of them. If this does not decrease the classification accuracy, the change is kept, otherwise it is rejected. Then the next pair of closest examples is considered. The algorithm works iteratively as long as there are no more examples to merge without affecting the classification accuracy.

2.2 RegENN, RegCNN and RegCA Instance Selection Algorithms for Regression Problems

Algorithm 1. RegENN algorithm

Require: $\mathbf{T}$
  $m \leftarrow sizeof(\mathbf{T})$;
  for $i = 1 \ldots m$ **do**
    $\bar{Y}(\mathbf{x}_i) =$NN$((\mathbf{T} \setminus \mathbf{x}_i), \mathbf{x}_i)$;
    $S \leftarrow$ Model$(\mathbf{T}, \mathbf{x}_i)$
    $\theta = \alpha \cdot std\,(Y\,(\mathbf{X}_S))$
    if $\left|Y(\mathbf{x}_i) - \bar{Y}(\mathbf{x}_i)\right| > \theta$ **then**
      $\mathbf{T} \leftarrow \mathbf{T} \setminus \mathbf{x}_i$
    end if
  end for
  $\mathbf{P} \leftarrow \mathbf{T}$
  return $\mathbf{P}$

Algorithm 2. RegCNN algorithm

Require: $\mathbf{T}$
  $m \leftarrow sizeof(\mathbf{T})$
  $\mathbf{P} = \emptyset$
  $\mathbf{P} \leftarrow \mathbf{P} \cup \mathbf{x}_1$;
  for $i = 2 \ldots m$ **do**
    $\bar{Y}(\mathbf{x}_i) =$NN$(\mathbf{P}, \mathbf{x}_i)$
    $S \leftarrow$ Model$(\mathbf{T}, \mathbf{x}_i)$
    $\theta = \alpha \cdot std\,(Y\,(\mathbf{X}_S))$
    if $\left|Y(\mathbf{x}_i) - \bar{Y}(\mathbf{x}_i)\right| > \theta$ **then**
      $\mathbf{P} \leftarrow \mathbf{P} \cup \mathbf{x}_i$;
      $\mathbf{T} \leftarrow \mathbf{T} \setminus \mathbf{x}_i$
    end if
  end for
  return $\mathbf{P}$

As the prediction algorithm for instance selection we use a weighted k-NN with $k = 9$, where the weight w_i exponentially decreases with the distance d_i between the given vector and its i-th neighbor x_i. The predicted output y is given by eq. 1.

$$y = \frac{\sum_{i=1}^{k} w_i y_i}{\sum_{i=1}^{k} w_i} \tag{1}$$

where $w_i = 2^{-0.2 d_i}$. We use Euclidean distance measure and a threshold Θ, which expresses the maximum difference between the output values of two vectors to consider them similar. Using Θ proportional to the standard deviation of k nearest neighbors of the vector x_i reflects the speed of changes of the output around xi and allows adjusting the threshold to that local landscape, what, as the experiments showed, allows for higher

compression of the dataset. As the regression model to predict the output Y(xi) we use k-NN with $k = 9$ as the Model(T,x_i) ($k = 9$ usually produced good results [15]). To adjust the CNN, ENN and CA algorithms to regression tasks the wrong/correct classification decision is replaced with a distance measure and a similarity threshold, to decide if a given vector can be considered as similar to its neighbors. Additionally in the CA algorithm, we define a regression counterpart of "the same class" by replacing it with "the output values being close enough". Thus, we have to find a pair of vectors, which are both: close in the input space (as in the original CA) and close in the output space. We merge these two vectors, for which the weighed sum D of the distances in the input and output space takes the smallest value:

$$D = d_{input} + \sqrt{num_{features}} \cdot d_{output} \tag{2}$$

The above formula requires the input and output values to be standardized. The distance in the output space is multiplied by the square root of the number of features to make the weight of it equal to the weight of the distance in the input space. If the priority is to obtain small dataset and thus make it easier to obtain simple logical rules from a regression tree built with the dataset, we can further modify the RegCA algorithm to allow for merging two closest vectors even if that slightly decreases the accuracy.

3 Regression Tree

A univariate regression tree is used with splitting of the data into nodes is based on variance minimization. The algorithms searches for all possible split points p of each feature f, as shown in the pseudo-code. The value optimized value is v and v_0 is the value at the previous iteration. Multiplying the variances of child nodes v_L, v_R by the number of vectors in the same nodes p_L, p_R causes that the splits are more symmetrical. Otherwise, it would be frequently only a few, or even one, vectors in one child node and all the other vectors in the other node. Such an unsymmetrical tree would have poor performance [16]. In our experiments the exponents n =1 and $m = 1$. To further improve the results, at the entrance of the tree the data can be transposed by a hyperbolic tangent to obtain rather uniform than Gaussian distribution [16]. One of the simplest forms of pruning is reduced error pruning. Starting at the leaves, each node is replaced with the average value of all vectors in the node and its subnodes. If the prediction accuracy does not decrease then the change is kept. There are two other possible ways to reduce the size of the tree. One of them is to use stopping criteria, such as minimal variance in the node. Once the minimal variance in the node is reached the node becomes a leaf. Another criterion that can be used together with the minimal variance is the minimal number of vectors in the node. If any node already reached that number of vectors if becomes a leaf. However, the problem with the stopping criteria is that their optimal value is unknown during building the tree and what is even more difficult, the optimal stopping criteria can be different for various nodes. Thus, the third discussed way to reduce the tree size is to reduce the training set size. If the training set is reduced with instance selection methods, the most representative instances remain in the dataset, allowing for building a smaller tree with better prediction ability. In regression problems there are several parameters that can be tuned in the RegENN, RegCNN and

RegCA algorithms. Depending on how the parameters are set more or fewer instances remain in the training set. If the parameters are set so that only few instances remain in the training set, then we can build the decision tree without any pruning or stopping criteria with a single instance in each leaf. Also any combinations are possible with either more intensive instance selection or more intensive pruning. The combinations are experimentally evaluated in the experimental section of the paper.

Algorithm 3. Tree optimization pseudo-code

Require: $\mathbf{F} = [f_1, f_2, \ldots, f_s]$
Ensure: $\underset{i=1:s}{\forall}\ \text{sizeof}(f_i) \leftarrow p$
  for $i = 1 \ldots s$ **do**
    $f_i = SortFeatureElements(f_i)$
    for $j = 1 \ldots p$ **do**
      $p_L = j/p$
      $p_R = (p - j)/p$
      $v = v_0 - p_L^m \cdot v_L^n - p_R^m \cdot v_R^n$
      if $v \geq q$ **then**
        $q = v$
        $s_0 = j$
        $f_0 = f_i$
      end if
    end for
  end for
  return s_0, f_0

4 Numerical Experiments

We implemented the instance selection algorithms in Java as RapidMiner Extensions and used RapidMiner [17] for the whole process. The decision tree was created in C# as a WPF application with graphical visualization and it is invoked in the ExecuteProgram module (fig. 1.) as well in the training as in the test part of the validation process. All the source codes, executable files and datasets used in the experiments can be downloaded from [18]. The whole process in different configurations was run in a 10-fold crossvalidation loop. Inside each crossvalidation run first he instance selection was performed and then the regression tree was created on the training dataset. The MSE was measured on the test dataset. The Training and Testing (the two lower windows in fig.1) constitute together a single run of the 10-fold crossvalidation (represented by the Validation module in the upper window in fig. 1). In the testing part, first the instance selection is made in various ways (in fig. 1. RegENN followed by RegCA is shown) then the selected examples (prototypes) are written to a CSV file and the ExecuteProgram module run the regression tree in the training mode as the external program. The tree reads the CSV dataset and builds the tree. In the Testing block, the tree is invoked from the ExecuteProgram module and it reads the structure of the decision tree (which was saved to a file in the Training block) and tests the tree on the test data, (the data was saved to a file by the WriteCSV(2) module). The ReadCSV(2) module reads the

prediction results of the tree and the Performance module calculates the MSE on the data. The Log module in the MainProcess block calculates the MSE and its standard deviation over the whole crossvalidation.

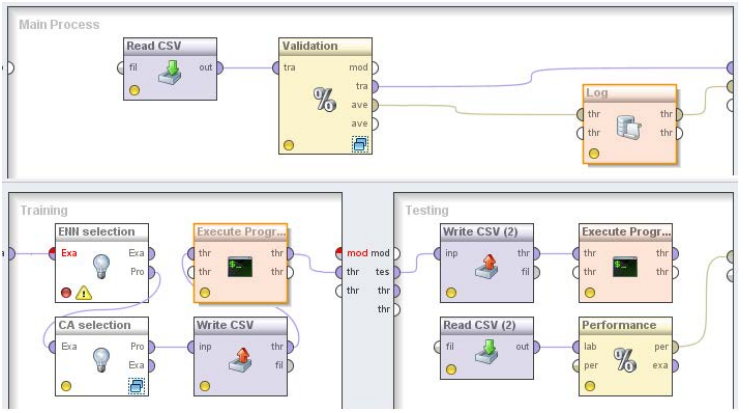


Fig. 1. The experimental process as implemented in Rapidminer

We performed the experiments on four datasets. First all the datasets were standardized so that the mean value of each attribute is zero and the standard deviation is one to make comparison of the results easy. Four datasets come the UCI Machine Learning Repository [19]: Concrete Compression Strength (7 attributes, 1030 instances), Crime and Communities (7 attr., 320 inst.), Housing (13 attr., 506 inst.). One dataset (Steel: 12 attr., 960 inst.) depicts the steel production process with the task to predict the amount of carbon that must be added to the liquid steel in to obtain desired steel properties. We experimentally evaluated the optimal Θ (see section IIB) and we used $\Theta = (5 \div 7) \cdot std_dev$ of 9 nearest neighbors for RegENN and $\Theta = (0.15 \div 0.25) \cdot std_dev$ of 9 nearest neighbors for RegCNN and RegCA. Additionally each time the node variance=0.001 was used as the stopping criteria.

Table 1. Experimental results for the Steel dataset

Regularization method	MSE	nodes	vectors
node variance	0.23±0.07	77±10	864±0
tree pruning	0.20±0.07	62±8	864±0
ENN	0.23±0.07	70±8	776±2
CNN	0.20±0.06	68±8	746±8
CA	0.18±0.05	62±8	701±4
ENN+CNN	0.18±0.05	62±7	722±10
ENN+CA	0.17±0.04	56±6	687±5
ENN+CA+pruning	0.16±0.04	53±6	687±5

Table 2. Experimental results for the Crime dataset

Regularization method	MSE	nodes	vectors
node variance	0.68±0.09	85±10	288±0
tree pruning	0.63±0.08	72±9	288±0
ENN	0.68±0.08	77±8	210±2
CNN	0.66±0.06	79±8	243±2
CA	0.63±0.06	72±8	202±3
ENN+CNN	0.64±0.07	72±9	197±4
ENN+CA	0.60±0.06	67±8	187±2
ENN+CA+pruning	0.58±0.06	63±8	187±2

Table 3. Experimental results for the Concrete dataset

Regularization method	MSE	nodes	vectors
node variance	0.88±0.08	167±26	927±0
tree pruning	0.85±0.07	125±22	927±0
ENN	0.92±0.08	72±9	277±5
CNN	0.96±0.09	142±22	786±6
CA	0.88±0.07	122±18	609±5
ENN+CNN	0.86±0.07	63±6	186±4
ENN+CA	0.83±0.06	56±6	180±4
ENN+CA+pruning	0.82±0.06	48±6	180±4

Table 4. Experimental results for the Housing dataset

Regularization method	MSE	nodes	vectors
node variance	0.41±0.08	105±14	455±0
tree pruning	0.39±0.08	68±11	455±0
ENN	0.43±0.09	70±10	350±4
CNN	0.39±0.08	68±10	387±3
CA	0.39±0.08	68±10	379±3
ENN+CNN	0.39±0.08	63±9	339±5
ENN+CA	0.36±0.06	63±9	319±4
ENN+CA+pruning	0.34±0.06	58±8	319±4

5 Conclusions

We presented an extension of CNN, ENN and CA algorithms, called RegCNN, RegENN and RegCA that can be applied to regression tasks and experimentally evaluated the influence of the Θ parameter on the number or selected vectors and the size and prediction accuracy of the regression tree. The best results were obtained when RegENN was used together with RegCA. It always improved the properties of the regression tree (smaller size and higher prediction accuracy), however after applying tree pruning in several cases the results could still be improved. Although the parameter Θ must be determined, the algorithms are not very sensitive to changes of the parameters and as rule of thumb it can be set to 6 for RegENN and to 0.2 for RegCNN and RegCA for

standardized datasets. In some cases we were able to obtain slightly better results using more complex algorithms (e.g. and MLP network), which were trained only on a part of the dataset situated closest to the vector of interest (local experts). However, that increases the time of instance selection by two or three orders of magnitude and make the process more complex than the k-NN. So far we adjusted to regression only some of the simplest instance selection methods and it would be worth to perform the study with other methods as well. Although the instance selection in regression problems does not reduce the dataset size so much as in classification tasks, it is worth performing not only to compress the data but also to improve the prediction of the model.

Acknowledgment. The work was sponsored by the grant No. ATH/2/IV/GW/2011 from the University of Bielsko-Biala.

References

1. Hart, P.E.: The condensed nearest neighbor rule. IEEE Transactions on Information Theory 14, 515–516 (1968)
2. Wilson, D.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics 2, 408–421 (1972)
3. Chang, C.L.: Finding prototypes for nearest neighbor classifiers. IEEE Transactions on Computers 23, 1179–1184 (1974)
4. Wilson, D., Martinez, T.: Reduction techniques for instance-based learning algorithms. Machine Learning 38, 251–268 (2000)
5. Cameron-Jones, R.M.: Instance selection by encoding length heuristic with random mutation hill climbing. In: The Eighth Australian Joint Conference on Artificial Intelligence, pp. 99–106 (1995)
6. Salvador, G., Derrac, J., Ramon, C.: Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 417–435 (2012)
7. Jankowski, N., Grochowski, M.: Comparison of instances seletion algorithms I. Algorithms survey. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 598–603. Springer, Heidelberg (2004)
8. Kordos, M., Blachnik, M., Wieczorek, T.: Temperature Prediction in Electric Arc Furnace ith Neural Network Tree. In: Honkela, T. (ed.) ICANN 2011, Part II. LNCS, vol. 6792, pp. 71–78. Springer, Heidelberg (2011)
9. Zhang, J., et al.: Intelligent selection of instances for prediction functions in lazy learning algorithms. Artifcial Intelligence Review 11, 175–191 (1997)
10. Tolvi, J.: Genetic algorithms for outlier detection and variable selection in linear regression models. Soft Computing 8, 527–533 (2004)
11. Guillen, A., et al.: Applying Mutual Information for Prototype or Instance Selection in Regression Problems. In: ESANN 2009 Proceedings (2009)
12. Duch, W., Setiono, R., Zurada, J.: Computational intelligence methods for understanding of data. Proceedings of the IEEE 92(5), 771–805 (2008)
13. Wu, S.: Optimal instance selection for decision tree, PhD dissertation. Iowa State university (2007)
14. Ramon Cano, J., Herrera, F., Lozano, M.: Evolutionary Stratified Training Set Selection for Extracting Classification Rules with Trade off Precision-Interpretability. Data and Knowledge Engineering 60, 90–108 (2006)

15. Kordos, M., Blachnik, M., Strzempa, D.: Do We Need Whatever More Than k-NN? In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part I. LNCS (LNAI), vol. 6113, pp. 414–421. Springer, Heidelberg (2010)
16. Kordos, M., Blachnik, M., Perzyk, M., Kozłowski, J., Bystrzycki, O., Gródek, M., Byrdziak, A., Motyka, Z.: A Hybrid System with Regression Trees in Steel-Making Process. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS, vol. 6678, pp. 222–230. Springer, Heidelberg (2011)
17. http://www.rapid-i.com
18. http://www.ath.bielsko.pl/~mkordos/icann2013
19. Merz, C., Murphy, P.: UCI repository of machine learning databases, http://www.ics.uci.edu/mlearn/MLRepository.html

On Perturbation Measure of Clusters: Application

Maciej Krawczak[1,2] and Grażyna Szkatuła[1]

[1] Systems Research Institute, Polish Academy of Sciences
Newelska 6, Warsaw, Poland
[2] Warsaw School of Information Technology
Newelska 6, Warsaw, Poland
{krawczak,szkatulg}@ibspan.waw.pl

Abstract. In this paper we developed a new methodology for grouping objects described by nominal attributes. We introduced a measure of perturbation of one cluster by another cluster in order to create a junction of clusters. The developed method is hierarchical and agglomerative and can be characterized both by high speed of computation as well as surprising good accuracy of clustering. keywords cluster analysis, nominal attributes, sets theory.

1 Introduction

Often there are collected data characterized by huge number of objects and the objects are characterized by a number of attributes, often of categorical (nominal) nature. However, there are algorithms specialized to analysis of long chains of symbols, the algorithms found applications in text analysis or in bioinformatics (Apostolico *et al.*, 2002; Gionis and Mannila, 2003; Lin *et al.*, 2007). Generally, algorithms dealing with symbolic data are based on introduction of some distance between objects, e.g. Wang (2010).

Our approach to cluster analysis with symbolic data differs from algorithms known in the literature and the efficiency of it seems to be higher than those known in literature. The developed algorithm has several features common with standard ones, namely our algorithm is hierarchical and agglomerative ("bottom-up") one. Hierarchical clustering (defined by Johnson in 1967) is starting with N clusters (each containing one object). This kind of algorithms can find the most similar pair of clusters and merge them into a single cluster. This kind of hierarchical clustering is called *agglomerative* because it merges clusters iteratively. The main weaknesses of agglomerative clustering methods are that they can never undo what was done previously. In our algorithm instead of measure of distance between objects we introduced a definition of the measure of perturbation of one cluster by another cluster – the approach allows merging smaller clusters in order to get larger ones.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 176–183, 2013.

2 Approach Details

First we will introduce several descriptions. Let us consider a finite set of objects $U = \{e_n\}$, $n = 1, 2, ..., N$. The objects are described in the form of conditions associated with the finite set of K attributes $A = \{a_1, ..., a_K\}$. The set $V_{a_j} = \{v_{j,1},\ v_{j,2}, ..., v_{j,L_j}\}$ represents the domain of the attribute $a_j \in A$, $j = 1, ..., K$, where L_j denotes the number of nominal values of the j-th attribute. This way each object $e_n \in U$ is represented by K sets in the following manner

$$e_n =< \{v_{1,t(1,n)}\}, \{v_{2,t(2,n)}\}, ..., \{v_{K,t(K,n)}\} > \tag{1}$$

where $v_{j,t(j,n)} \in V_{a_j}$ and $j = 1, ..., K$. The used index $t(j, n)$ for $j \in \{1, 2, ..., K\}$ and $n \in \{1, 2, ..., N\}$ denotes that the attribute a_j takes the value $v_{j,t(j,n)}$ in the object e_n.

For instance, for the attribute a_j and $L_j = 4$, using letters of the alphabet, the set V_{a_j} can have the following nominal form $V_{a_j} = \{a,\ b,\ c,\ d\}$. An exemplary data object for a given $n \in \{1, ..., N\}$ and $K = 4$ can be written as follows:

$$e_n =< \{b\}, \{d\}, \{a\}, \{c\} > .$$

The aim of clustering can be formulated in the following way: we want to split the set of objects U into non-empty, disjoint subsets (called *clusters*) $G_{g_1},\ G_{g_2}, ..., G_{g_C}$, subject to $\bigcup_{i=1}^{C} G_{g_i} = U$ and $G_{g_u} \cap G_{g_w} = \emptyset$, for $u \neq w$, in such a way that objects in each cluster are 'similar' in some sense, and the objects from different clusters should be 'dissimilar'. The set of clusters on U is denoted by $G(U)$.

In the new proposed method *a measure of clusters' perturbation* is introduced, and it describes in some sense clusters' similarity and/or clusters' dissimilarity. The proposed algorithm belongs to a family of hierarchical clustering algorithms.

The procedure starts with N objects as individual clusters and progresses up to finding the whole set U as one cluster. A pair of clusters described by the lowest value of measure of perturbation is coupled and in such a way a new cluster is formed - this way the number of clusters is decreased by one. The progress of the procedure is stopped when a fixed number of clusters C, $C < N$, is found. It is assumed, that if a certain object belongs to a defined cluster then the same object must not be included into another cluster. There are basic elements of proposed method introduced below.

At the beginning we will use a term *group* instead of cluster. Namely, every non empty subset of a finite set of objects U is called a *group*. Every group g, $g \subseteq U$, can be represented by an ordered collection of K elements – it means sets of values of the attributes describing objects,

$$G_g =< A_{1,t(1,g)},\ A_{2,t(2,g)}, ..., A_{K,t(K,g)} >, \tag{2}$$

where $A_{j,t(j,g)} \subseteq V_{a_j}$, $card(A_{j,t(j,g)}) \geqslant 1$ for $j \in \{1,\ ...\ , K\}$.

Simple example will illustrate (2), for instance**,** for three symbolic attributes $\{a_1, a_2, a_3\}$, each having the following domains $V_{a_1} = \{a, b, c\}$, $V_{a_2} = \{d, e\}$,

$V_{a_3} = \{f, h, n\}$, we can describe as a group g, e.g. by $< \{a, c\}, \{d\}, \{f, n\} >$ or by $< \{a\}, \{d\}, \{n\} >$.

Now, let us consider an attribute a_j and the sets of attributes values $A_{j,\,t(j,\,g_1)}$ and $A_{j,\,t(j,\,g_2)}$, where $A_{j,\,t(j,\,g_1)} \subseteq V_{a_j}$, $A_{j,\,t(j,\,g_2)} \subseteq V_{a_j}$, for a fixed j. The idea of perturbation of one group by another is as follows, if we attach the first set to the second set then such an action can be considered that the second set is perturbed by the first set - in other words the set $A_{j,t(j,g_1)}$ perturbs the set $A_{j,t(j,g_2)}$. Here we propose the following way to measure a level of perturbation one set by another.

Definition 1. *Measure of perturbation of a set $A_{j,t(j,g_2)}$ by a set $A_{j,t(j,g_1)}$ is defined in the following manner:*

$$Per(A_{j,t(j,g_1)} \mapsto A_{j,t(j,g_2)}) = \frac{card(A_{j,t(j,g_1)} \backslash A_{j,t(j,g_2)})}{card(V_j) - 1} \ . \tag{3}$$

Assuming that the following set $V_{a_1} = \{a,\, b,\, c,\, d,\, e\}$ is considered, here there are a few exemplary measures of perturbation of two subsets $A_{1,t(1,g_1)}$ and $A_{1,t(1,g_2)}$ of the set V_{a_1}.

For $A_{1,t(1,g_1)} = \{a,\, b,\, c\}$ and $A_{1,t(1,g_2)} = \{d, e\}$ using the definition (3) we obtain the following measures of perturbation of one set by another, and vice versa:

$$Per(A_{1,t(1,g_1)} \mapsto A_{1,t(1,g_2)}) = 3/4,\ Per(A_{1,t(1,g_2)} \mapsto A_{1,t(1,g_1)}) = 2/4,$$

and for another pair of sets $A_{1,t(1,g_1)} = \{a,\, b, c, d, e\}$ and $A_{1,t(1,g_2)} = \{e\}$, we obtain as follows

$$Per(A_{1,t(1,g_1)} \mapsto A_{1,t(1,g_2)}) \quad = 1,\ Per(A_{1,t(1,g_2)} \mapsto A_{1,t(1,g_1)}) \quad = 0.$$

Now we will introduce another definition:

Definition 2. *Measure of perturbation of G_{g_2} by G_{g_1} (denoted $Per(G_{g_1} \mapsto G_{g_2})$) is defined in the following manner:*

$$Per(G_{g_1} \mapsto G_{g_2}) = \frac{1}{K} \sum_{j=1}^{K} Per(A_{j,t(j,g_1)} \mapsto A_{j,t(j,g_2)}). \tag{4}$$

It is easy to notice that (4) can be rewritten as follows

$$Per(G_{g_1} \mapsto G_{g_2}) = \frac{1}{K} \sum_{j=1}^{K} \frac{card(A_{j,t(j,g_1)} \backslash A_{j,t(j,g_2)})}{card(V_j) - 1} \ . \tag{5}$$

The measure of perturbation is assumed to return a value from [0, 1], where 1 is interpreted as most level for perturbation, while 0 is the lowest level for perturbation. It should be noticed that this measure is asymmetrical, so this measure cannot be considered as the distance between the groups.

Now let us consider a pair of groups g_1 and g_2 described as follows: $G_{g_1} =< A_{1,\,t(1,\,g_1)},\ \ ...,\, A_{K,\,t(K,\,g_1)} >$ and $G_{g_2} =< A_{1,\,t(1,\,g_2)},\ \ ...,\, A_{K,\,t(K,\,g_2)} >$, where $A_{j,\,t(j,g_1)} \subseteq V_{a_j}$, $A_{j,\,t(j,g_2)} \subseteq V_{a_j}$, $j \in \{1,\, 2,\, ...\,,\, K\}$. The group $g_1$contains the objects $\{e_n : n \in J_{g_1} \subseteq \{1,\, ...,\, N\}\}$, and g_2 contains the objects $\{e_n : \ \ n \in J_{g_2} \subseteq \{1,\, ...,\, N\}\}$, where $J_{g_1} \cap J_{g_2} = \emptyset$. The *join between* these *groups* is described as:

$$G_{g_1} \oplus G_{g_2} =< A_{1,\,t(1,\,g_1)} \cup A_{1,\,t(1,\,g_2)},\ A_{2,\,t(2,\,g_1)} \cup A_{2,\,t(2,\,g_2)},\ ...,\, A_{K,\,t(K,\,g_1)} \cup A_{K,\,t(K,\,g_2)} > . \tag{6}$$

Created a new group g_3, $G_{g_3} := G_{g_1} \oplus G_{g_2}$**,** contains the following objects $\{e_n : n \in J_{g_1} \cup J_{g_2}\}$. The way of construction of a new group C_{g_3} is shown in Table 1.

Table 1.

Group \ Attribute	a_1	a_2	a_3
G_{g_1} : $e1$, $e2$	e	e	$g \vee h$
G_{g_2} : $e3$, $e4$, $e5$	f	e	g
G_{g_3} : $e1$,$e2$, $e3$, $e4$, $e5$	$e \vee f$	e	$g \vee h$

The new group $G_{g_3} =< \{e, f\},\, \{e\},\, \{g, h\} >$contains objects $e1$, $e2$, $e3$, $e4$ and $e5$.

In the succeeding part of the paper we will apply Definition 1 as well as Definition 2 for clusters as groups of objects.

3 Clusters Merging Algorithm

3.1 The Algorithm

In this paper we propose a hierarchical agglomerative approach to cluster nominal data. The bottom level of the structure has singular clusters with separate objects while the top level contains one cluster with all objects. During iteration two clusters are heuristically selected. These selected clusters are then merged to form a new cluster.

Assuming: U - set of objects, $card(U) = N$, K - number of attributes, C - assumed number of clusters. The proposed algorithm is formulated as follows:
Step 1.
Each of N example creates one-element cluster in the initial set of clusters $G\,(U)$, $card(G(U)) = N$, i.e. $G(U) = \{G_{g_1},\, G_{g_2}, ..., G_{g_N}\}$, iteration $k = 0$.
Step 2.
Iteration $k = k + 1$. Create a matrix *MPer* of measures of increasing of the cluster,

MPer: $card(G(U)) \times card(G(U))$, where $MPer[p, q] := Per(G_{g_p} \mapsto G_{g_q})$,

$$p = 1,\, 2, ...,\, card(G(U)), \quad q = 1, 2, ..., card(G(U)), \quad p \neq q.$$

Step 3.
Find two clusters $G_{g_{p*}}$ and $G_{g_{q*}}$ that minimize following criterion:

$$Per\ (G_{g_{p*}} \mapsto G_{g_{q*}}) := \min_{\substack{\forall p \in \{1,2,...,card(G(U)\} \\ \forall q \in \{1,2,...,card(G(U)\} \\ p \neq q}} Per(G_{g_p} \mapsto G_{g_q}).$$

Step 4.
Create a new cluster in the set of clusters $G(U)$, $G_{g_{p*,q*}} := G_{g_{p*}} \oplus G_{g_{q*}}$. The clusters $G_{g_{p*}}$ and $G_{g_{q*}}$ are removed from the set $G(U)$. Thus, $card(G(U)) := card(G(U)) - 1$.
Step 5.
If the required number $card(G(U)) = C$ is reached, STOP; otherwise, return to Step 2 and modify the matrix *MPer*.

The modification of $MPer[p, q]$ relies on removing of the $p*$-th and $q*$-th rows as well as the $p*$-th and $q*$-th columns and at the end adding a new row and column. The new row and column are related to the new cluster $G_{g_{p*,q*}}$. The measures $Per(G_{g_{p*,q*}} \mapsto G_{g_j})$ for $j = 1, \ldots, card(G(U)) - 1$ and $Per(G_{g_j} \mapsto G_{g_{p*,q*}})$ for $i = 1, \ldots, card(G(U)) - 1$ are counted.

This way, the disjoint set of clusters $G(U) = \{G_{g_1}, G_{g_2}, ..., G_{g_C}\}$, where $card(G(U))$, determines the set of clusters with assumed cardinality.

3.2 Illustrating Example

Let us consider a set of objects described in Table 2.

Table 2.

Object \ Attribute	a_1	a_2	a_3	a_4	a_5
$e1$	c	b	a	a	b
$e2$	b	a	b	a	c
$e3$	d	b	c	a	b
$e4$	d	a	a	b	a
$e5$	b	a	b	b	a
$e6$	d	b	c	a	b

All objects $e1, e2, e3, e4, e5, e6$ are described in the form of conditions associated with the set of attributes $\{a_1, \ldots, a_5\}$. The set V_{a_j} describes the domain of attribute a_j, $j = 1, ..., 5$; then we have the following attributes domains $V_{a_1} = \{b, c, d\}$, $V_{a_2} = \{a, b\}$, $V_{a_3} = \{a, b, c\}$, $V_{a_4} = \{a, b\}$, $V_{a_5} = \{a, b, c\}$.

Our aim is to group the objects into prescribed number of two clusters, $C = 2$. At the beginning we assume that each object creates one-element cluster in the initial set of clusters $G(U) = \{G_{g_1}, G_{g_2}, ..., G_{g_6}\}$, $card(G(U)) = 6$, in the following way:

$$G_{g_1} =< \{c\}, \{b\}, \{a\}, \{a\}, \{b\} >, \ldots, G_{g_6} =< \{d\}, \{b\}, c\}, \{a\}, \{b\} > .$$

We calculate values of measure of perturbations of cluster G_{g_j} by G_{g_i} (in a form of a matrix *MPer*), where $MPer[i,j] = Per(G_{g_i} \mapsto G_{g_j})$, $i \neq j, i = 1, 2, \ldots, 6$, $j = 1, 2, \ldots, 6$, see Table 3.

Table 3. Matrix *MPer*

Cluster \ Cluster	G_{g_1}	G_{g_2}	G_{g_3}	G_{g_4}	G_{g_5}	G_{g_6}
G_{g_1}	-	1/2	1/5	3/5	7/10	2/10
G_{g_2}	1/2	-	1/2	1/2	3/10	1/2
G_{g_3}	1/5	1/2	-	3/5	7/10	0
G_{g_4}	3/5	1/2	3/5	-	1/5	3/5
G_{g_5}	7/10	3/10	7/10	1/5	-	7/10
G_{g_6}	2/10	1/2	0	3/5	7/10	-

The minimal values in Table 3 appear for two clusters G_{g_3} and G_{g_6}, then from a pair of clusters G_{g_3} and G_{g_6} a new cluster G_{g_7} is created, while clusters G_{g_3} and G_{g_6} are removed from the set $G(U)$. Thus, $card(G(U)) := card(G(U)) - 1 = 5$, see Table 4. The new calculations are shaded in the forthcoming tables.

Table 4.

Cluster \ Attribute	a_1	a_2	a_3	a_4	a_5
G_{g_1} : $\{e_1\}$	c	b	a	a	b
G_{g_2} : $\{e_2\}$	b	a	b	a	c
G_{g_4} : $\{e_4\}$	d	a	a	b	a
G_{g_5} : $\{e_5\}$	b	a	b	b	a
G_{g_7} : $\{e_3, e_6\}$	d	b c		a	b

Because the number $card(G(U)) = 5 > 2$, thus we modify the table of cluster's perturbations *MPer* in a way shown in Table 5.

The new cluster G_{g_8} is created on the base of G_{g_1} and G_{g_7}, see Table 6.

Due to the number $card(G(U)) = 4 > 2$, we can modify the last table of cluster's perturbations in the following way shown in Table 7.

From the pair of clusters G_{g_4} and G_{g_5} the new cluster G_{g_9} is arranged, see Table 8.

On the ground of $card(G(U)) = 3 > 2$, thus again we modify the table of cluster's perturbations, see Table 9.

The pair of clusters G_{g_2} and G_{g_9} constitute a new cluster $G_{g_{10}}$, see Table 10.

This way the required number of clusters has been already reached, $card(G(U)) = 2$, and we were able to obtain the following set of clusters $G(U) = \{G_{g_8}, G_{g_{10}}\}$, where the first cluster is described in the following way

Table 5.

Cluster \ Cluster	G_{g_1}	G_{g_2}	G_{g_4}	G_{g_5}	G_{g_7}
G_{g_1}	-	1/2	3/5	7/10	1/5
G_{g_2}	1/2	-	1/2	3/10	1/2
G_{g_4}	3/5	1/2	-	1/5	3/5
G_{g_5}	7/10	3/10	1/5	-	7/10
G_{g_7}	1/5	1/2	3/5	7/10	-

Table 6.

Cluster \ Attribute	a_1	a_2	a_3	a_4	a_5
G_{g_2} : $\{e_2\}$	b	a	b	a	c
G_{g_4} : $\{e_4\}$	d	a	a	b	a
G_{g_5} : $\{e_5\}$	b	a	b	b	a
G_{g_8} : $\{e_1, e_3, e_6\}$	c, d	b	a, c	a	b

Table 7.

Cluster \ Cluster	G_{g_2}	G_{g_4}	G_{g_5}	G_{g_8}
G_{g_2}	-	1/2	3/10	1/2
G_{g_4}	1/2	-	2/10	1/2
G_{g_5}	3/10	2/10	-	7/10
G_{g_8}	7/10	7/10	9/10	-

Table 8.

Cluster \ Attribute	a_1	a_2	a_3	a_4	a_5
G_{g_2} : $\{e_2\}$	b	a	b	a	c
G_{g_8} : $\{e_1, e_3, e_6\}$	c, d	b	a, c	a	b
G_{g_9} : $\{e_4, e_5\}$	b, d	a	a, b	b	a

Table 9.

Cluster \ Cluster	G_{g_2}	G_{g_8}	G_{g_9}
G_{g_2}	-	1/2	3/10
G_{g_8}	7/10	-	7/10
G_{g_9}	1/2	7/10	-

Table 10.

Cluster \ Attribute	a_1	a_2	a_3	a_4	a_5
G_{g_8} : $\{e_1, e_3, e_6\}$	c, d	b	a, c	a	b
$G_{g_{10}}$: $\{e_2, e_4, e_5\}$	b, d	a	a, b	a, b	a, c

$$G_{g_8} = < \{c, d\},\ \{b\},\ \{a, c\},\ \{a\},\ \{b\} > \text{and } G_{g_8}: \{e_1,\ e_3,\ e_6\},$$

while the second cluster has the following description

$$G_{g_{10}} = < \{b, d\},\ \{a\},\ \{a, b\},\ \{a, b\},\ \{a, c\} > \text{and } G_{g_{10}}: \{e_2,\ e_4,\ e_5\}.$$

4 Conclusions

In this paper we described a new approach for building clusters of objects described by nominal data. For such objects' description we introduced and developed the new algorithm based on the idea of perturbation of one cluster by another, or perturbation of conditions of each pair of clusters. The introduced measure of perturbation of two clusters allowed us to choose a pair of the most 'similar' clusters in order to join them and create a new one. It should be emphesised that the measure of clusters' perturbation is asymmetrical. The solved example shows the efficiency of the new developed method for grouping objects described by nominal attributes, which seems to be both effective as well as elegant.

References

1. Apostolico, R., Bock, M.E., Lonardi, S.: Monotony of surprise in large-scale quest for unusual words. In: Proceedings of the 6th International Conference on Research in Computational Molecular Biology, Washington, DC, April 18-21, pp. 22–31 (2002)
2. Gionis, A., Mannila, H.: Finding recurrent sources in sequences. In: Proceedings of the 7th International Conference on Research in Principles of Database Systems, Tucson, AZ, May 12-14, pp. 249–256 (2003)
3. Johnson, S.C.: Hierarchical Clustering Schemes. Psychometrika 2, 241–254 (1967)
4. Krawczak, M., Szkatuła, G.: On time series envelopes for classification problem. In: Developments of Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets, vol. II (2010)
5. Krawczak, M.: Szkatuła G, Time series envelopes for classification. In: Proceedings of the Conference: IEEE International Conference on Intelligent Systems, London, UK, July 7-9, pp. 156–161 (2010)
6. Krawczak, M., Szkatuła, G.: A hybrid approach to dimension reduction in classification. Control and Cybernetics 1(2), 527–551 (2011)
7. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing SAX: a Novel Symbolic Representation of Time Series. Data Min Knowledge Disc 2(15), 107–144 (2007)
8. Wang, B.: A New Clustering Algorithm on Nominal Data Sets. In: Proceedings of International MultiConference of Engineers and Computer Scientists 2010 IMECS 2010, Hong Kong, March 17-19 (2010)

Using Topology Preservation Measures for Multidimensional Intelligent Data Analysis in the Reduced Feature Space

Szymon Łukasik[1,2] and Piotr Kulczycki[1,2]

[1] Systems Research Institute, Polish Academy of Sciences,
ul. Newelska 6, 01-447 Warsaw, Poland
[2] Department of Automatic Control and Information Technology,
Cracow University of Technology
ul. Warszawska 24, 31-155 Cracow, Poland
slukasik@ibspan.waw.pl

Abstract. This paper investigates a possibility of supplementing standard dimensionality reduction procedures, used in the process of knowledge extraction from multidimensional datasets, with topology preservation measures. This approach is based on an observation that not all elements of an initial dataset are equally preserved in its low-dimensional embedding space representation. The contribution first overviews existing topology preservation measures, then their inclusion in the classical methods of exploratory data analysis is being discussed. Finally, some illustrative examples of presented approach in the tasks of cluster analysis and classification are being given.

Keywords: multidimensional datasets, dimensionality reduction, topology preservation, cluster analysis, classification.

1 Introduction

Recently, the subject of intelligent data analysis are predominantly high dimensional datasets with huge sample lengths. It is a result of growing amount of information stored in distributed data warehouses and fast development of data-intensive software applications frameworks [3]. The knowledge extraction and visualization of such datasets are difficult, mainly due to methodological obstacles of high dimensional data analysis. They are caused mainly by inherent properties of such datasets referred in bibliography as "curse of dimensionality" [16].

To overcome those issues numerous dimensionality reduction procedures have been proposed. Let X to denote $n \times m$ data matrix:

$$X = \left[x_1 \; x_2 \; ... \; x_m \right] \tag{1}$$

columns of which represent n dimensional sample elements for given real-valued probabilistic variable. Each dimension of such variable will be referred later in

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 184–193, 2013.

this paper as a feature. The general aim of dimensionality reduction is a data transformation to its new $N \times m$ sized form, where N is significantly smaller than n. This can be achieved either by selecting most N significant features (feature selection) or by construction of a new set of N features (feature extraction) based on the initial ones. The second case is more general and will be under consideration here. Among feature extraction procedures one can distinct: linear methods where synthesis of resulting dataset Y is performed by linear transformation:

$$Y = A\,X \tag{2}$$

with A being a transformation matrix of size $N \times n$ and nonlinear techniques where data transformation can be described by a nonlinear function $g : R^n \to R^N$ (or if such functional relationship does not exist). Detailed study on performance of routines belonging to both of above mentioned classes can be found in [10].

The general goal of dimensionality reduction is removing dataset's redundant content, however at the same time its application can cause a loss of important information carried within its entries. The latter can be quantitatively evaluated using different preservation quality indices, measuring datasets structural deformation. Some of those indicators can be considered on per-element basis which directly allows to assess how well each element of the dataset was relatively preserved by the dimensionality reduction transformation. Such approach is investigated in this paper, along with a novel concept of using this index (named below as a elements weight) to improve the performance of intelligent data analysis procedures in the reduced feature space. The presented idea was suggested first in our previous contribution devoted to the novel, metaheuristic-based dimensionality reduction technique [8], as well as in [9], where it was experimentally assessed for raw stress measure.

The paper is organized as follows. Various topology preservation indices already presented in the bibliography of the subject are given in the following Section. The use of some of them, on per-element basis, for selected data analysis procedures in the reduced feature space is discussed in Section 3, along with experimental results given in Section 4. Finally, the last part of the contribution contains some concluding remarks on the introduced approach and planned further research.

2 Topology Preservation Measures

Let us, in equivalence to (1), consequently denote the representation of given dataset in the reduced feature space by $N \times m$ data matrix:

$$Y = \left[y_1\ y_2\ ...\ y_m \right] \tag{3}$$

For the purpose of subsequent analysis we also define Euclidean distances between two datasets elements i and j ($i, j \in \{1, 2, ..., m\}$) in the initial and reduced feature space (d_{ij} and δ_{ij} accordingly) as follows:

$$d_{ij} = \|x_i - x_j\|_{R^n} \tag{4}$$

$$\delta_{ij} = \|y_i - y_j\|_{R^N} \tag{5}$$

Dimensionality reduction procedures are often classified into two, not always clearly distinguished, groups, namely global and local techniques [14]. The former are characterized by an attempt to preserve global geometrical properties of the original data in its low-dimensional representation Y, while the latter are based on trying to keep the local neighbourhood relations found initially in X.

To measure the quality of the global-based mapping one can use simple raw stress used in many variants of Multidimensional Scaling [1]:

$$S_R = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} (d_{ij} - \delta_{ij})^2 \tag{6}$$

as well as its normalized form provided by Sammon [12]

$$S_S = \frac{1}{\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} d_{ij}} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \frac{(d_{ij} - \delta_{ij})^2}{d_{ij}} \tag{7}$$

which puts less emphasis on large distances.

Stress-based indices are the most commonly used, however in many practical problems it is sufficient to evaluate the preservation of distances order rather than their exact values. Spearman's rho [13] can be employed in that case as it estimates the correlation of rank order data. In the context of dimensionality reduction, this coefficient can indicate how well the corresponding low-dimensional embedding preserves the order of pairwise distances between the original data points converted to ranks. Spearman's rho is calculated by using the following equation:

$$\rho_{SP} = 1 - \frac{6 \sum_{p=1}^{M} (r_{p_d} - r_{p_\delta})^2}{M^3 - M} \tag{8}$$

where $M = m(m-1)/2$ is a total number of distances subjected to the comparison and r_{p_d} , r_{p_δ} are the ranks (with $p = 1, 2, ..., M$) of pairwise distances sorted in ascending order for both, initial and reduced feature space. Spearman's rho value equal to 1 is equivalent to perfect preservation of distances' order (in general $\rho_{SP} \in [-1, 1]$).

Local mappings are usually evaluated using neighbourhood graph preservation. While, there exists numerous methods to analyze it e.g. Konig measure [6], simple one-parameter Mean Relative Rank Error (MRRE) index [7] will be presented here.

Let $\mathcal{N}_k(x_i)$ to represent a group of k-nearest neighbors of x_i, and $R^i_{j_d}$, $R^i_{j_\delta}$ be the ordered rank of distances d_{ij} and δ_{ij} respectively, defined for a set of all distances between element i and a rest of the dataset. MRRE is then defined as follows:

$$MRRE = \frac{1}{C} \sum_{i=1}^{m} \sum_{x_j \in \mathcal{N}_k(x_i)} \frac{|R^i_{j_d} - R^i_{j_\delta}|}{R^i_{j_d}} \tag{9}$$

with the following normalizing factor C:

$$C = m \sum_{p=1}^{k} \frac{|2p - m - 1|}{p} \tag{10}$$

which ensures that $MRRE \in [0, 1]$. This measure can be found similar to continuity, and it vanishes to zero if nearest neighbours of each data point appear in the same order in both spaces [7]. In this initial study we consider MRRE with $k = 11$.

For more detailed description and comparison of above mentioned measures one could refer to [4]. The next Section of this contribution will be devoted to the use of some of them in data analysis procedures performed in the reduced feature space.

3 Proposed Approach

Dimensionality reduction in general can significantly modify some data elements' relative position. Consequently the performance of data mining procedures defined in the reduced feature space can be seriously affected. Thus, it would be useful to synthesize individual measure which could serve as a quantitative index of how well each point of the dataset was relatively preserved by the dimensionality reduction transformation. This index element's weight w_i could be then used for exploratory data analysis procedures in the reduced feature space. Investigating if weights in such form are beneficial for data mining procedures performed in the space with reduced dimensionality constitutes one of the goals of this study.

To define the weight for each dataset element it is crucial first to directly evaluate a contribution w_i^* this elements embedding brings to the selected topology preservation index (note that these auxiliary coefficients do not have to sum up to the value of general index). Given the form of already introduced topology preservation measures these per-element factors could be defined accordingly, for raw stress:

$$w_i^* = S_{R_i} = \sum_{j=1}^{m} (d_{ij} - \delta_{ij})^2 \tag{11}$$

Sammon stress:

$$w_i^* = S_{S_i} = \frac{1}{\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} d_{ij}} \sum_{j=1}^{m} \frac{(d_{ij} - \delta_{ij})^2}{d_{ij}} \tag{12}$$

Spearmans rho (this time with $r_{p_d}^i$ and $r_{p_\delta^i}$ representing ranks of distances from p to the element i):

$$w_i^* = 1 - \rho_{SP_i} = \frac{6 \sum_{p=1}^{m} (r_{p_d}^i - r_{p_\delta}^i)^2}{M^3 - M} \tag{13}$$

and Mean Relative Rank Error:

$$w_i^* = MRRE_i = \frac{1}{C} \sum_{x_j \in \mathcal{N}_k(x_i)} \frac{|R_{j_d}^i - R_{j_\delta}^i|}{R_{j_d}^i} \tag{14}$$

Consequently weights are to be calculated using w_i^* values obtained from (11-14) and performing additional normalization:

$$w_i = \frac{m(w_i^*)^{-1}}{\sum_{i=1}^m (w_i^*)^{-1}} \tag{15}$$

for $i = 1, ..., m$ to ensure that

$$\sum_{i=1}^m w_i = m \ . \tag{16}$$

Please note that if $w_i^* = 0$ one should replace it with $\min_{j=1,...,m} w_j \neq 0$.

Introduction of weights allows to take into account deformations in a dataset relative structure. Data elements with higher weight might then be treated as more adequate. Besides using weights values directly one can use them also to eliminate the influence of some badly deformed data elements. It can be performed by neglecting in weight-based data analysis procedures by setting w_i to 0 those elements for which associated weights fulfil the following condition: $w_i < W$, where $W \in \mathbb{R}^+$ can be referred to as elimination threshold. Then all other weights should be either normalized to keep (16) or alternatively set to 1. This second variant of proposed approach will be under investigation here. The following two subsections of the paper will discuss how the general weight-based scheme defined above can be utilized for two standard data mining algorithms: clustering with K-means procedure and nearest neighbour classification.

3.1 Use Case 1: K-Means Clustering Algorithm

The task of cluster analysis is equivalent to such division of available data elements into subgroups (clusters) that elements belonging to each cluster are similar to each other and on the other hand there exist a significant dissimilarity between different clusters elements. The technique of data clustering considered here is based on a modification of the classic K-means algorithm. K-means is an iterative clustering algorithm which is aimed at minimizing sum-of-squares error i.e. sum of distances of dataset elements to their nearest cluster center $C_i = [c_1, c_2, ..., c_N]$, with $i \in 1, 2, ...K$. The procedure, in its standard form, includes a step of cluster assignment followed by clusters centers update [2]. The influence of topology preservation ratio in the reduced feature space can be included in the second stage of clustering algorithm. Each cluster center is then established using the following modified equation:

$$c_{ij} = \frac{1}{\sum_{y_l \in C_i} w_l} \sum_{y_l \in C_i} w_l y_{l_j} \ , \tag{17}$$

with $i = 1, ..., K$ and $j = 1, ..., N$. Consequently, such modified algorithm can be referred to as weighted K-means [5].

3.2 Use Case 2: Nearest Neighbour Classifier

Now let us consider the task of classification, that is designating element $\tilde{x} \in \mathbb{R}^n$ to one of the fixed class with known set of representative patterns (training set), similar to (1). Nearest neighbour classifier is a basic solution for this problem. The algorithm itself assigns element $\tilde{x}$ to a class which nearest neighbour of $\tilde{x}$ from the training set belongs to. Its modified variant, taking into account topology preservation, makes a similar decision on a basis of weighted distances i.e. divided additionally by weight w_i. This approach can be easily generalized for broader category of k-Nearest Neighbour Classifiers [11].

4 Experimental Results

Proposed technique was preliminarily verified for data exploration procedures performed for five multidimensional datasets taken from the UCI Machine Learning Repository [15] listed in Table 1.

Table 1. Used datasets description

Dataset	m	n	N	Classes	Class Description	Sample length
glass	214	9	4	6	*building_windows_float_processed*	70
					building_windows_non_float_processed	76
					vehicle_windows_float_processed	17
					containers	13
					tableware	9
					headlamps	29
wine	178	13	5	3	*producer_1*	59
					producer_2	72
					producer_3	47
WBC	683	9	4	2	*benign*	444
					malign	239
vehicle	846	18	10	4	*Opel*	212
					Saab	217
					bus	218
					van	199
seeds	210	7	2	3	*Kama*	70
					Rosa	70
					Canadian	70

Dimensionality reduction was performed using Principal Components Analysis. We used fixed values of embedding dimension N established in previous

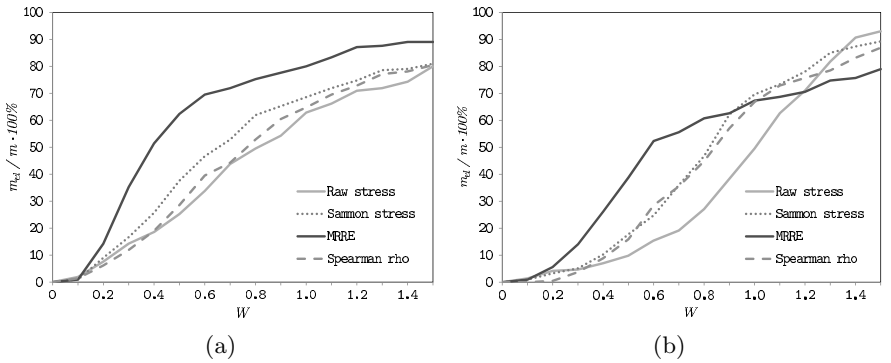


Fig. 1. Weights values distribution for *seeds* (a) and *glass* (b) datasets

experiments. The accuracy of K-means clustering was measured using Rand index value I_C calculated versus class labels, whereas for nearest-neighbour classification average classifier accuracy I_K during 5-fold cross validation was under close scrutiny. All experiments involving aforementioned data mining procedures where repeated 30 times, with mean and standard deviation being reported here (in "mean ± standard deviation" notation).

The initial experiments were conducted to evaluate the distribution of weight values calculated from (11-14). It was computationally verified by setting $W = 0.1, 0.2, ..., 1.5$ and observing the percentage (relative to the sample size m) of dataset elements with weight values under W, labelled as m_{el}. The results of those studies are shown on Figure 1. It can be seen that the distribution of weight values is not linear. However, for all considered datasets less than 50% of sample elements are characterized by weights values below average i.e. $w_i < 1$. It was also observed that weighting scheme based on raw stress could tend to be conservative, whereas using MRRE would offer neglecting large part of the dataset even for a small value of W.

The next series of experiments was designed to investigate the performance improvement for clustering, when using weighted variant of K-means algorithm with varying values of threshold W. First, the standard K-means procedure was tested for the reduced dataset, with $I_c \cdot 100\%$ representing the Rand index value obtained at that stage. Then the algorithm supported by aforementioned weighting scheme was executed with multiple runs measuring the performance variation $\Delta I = (I_c - I_{c_W}) \cdot 100\%$ for different values of W. Figure 2 exhibits the results obtained at that stage for two selected datasets and different weighting schemes.

For all datasets introducing the weights based on topology preservation index improve clustering performance. For most cases it is also recommended to neglect some dataset elements here. Increasing W beyond 1 however, leads to spectacular decrease in clustering accuracy.

The summary of our results, concerning both cluster analysis and classification is given in Table 2. Along with standard data mining procedures their modified

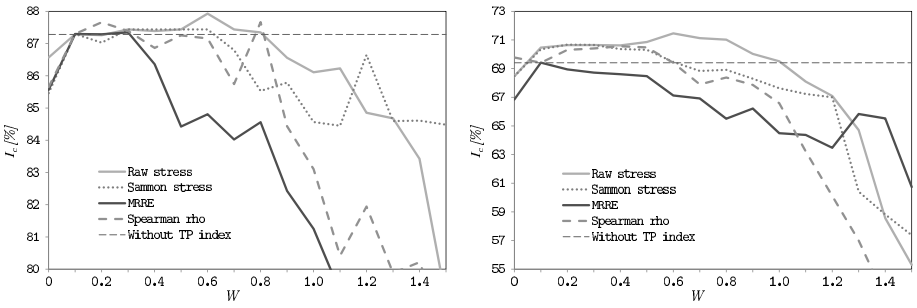


Fig. 2. Comparison of Rand indices obtained using K-means in reduced feature space with, and without topology preservation index weights for *seeds* (a) and *glass* (b) datasets

Table 2. Clustering and classification in reduced feature space - comparison of standard approach and our proposal ($I_c \cdot 100\%$ for clustering and I_k for classification were reported)

Procedure	***glass***	***wine***	***WBC***	***vehicle***	***seeds***
PCA, K-means	69.41 ±2.43	71.37 ±1.07	92.51 ±0.13	64.21 ±1.91	87.28 ±0.15
PCA, weighted K-means	71.46 ±1.88	71.40 ±0.98	93.21 ±0.00	64.21 ±1.80	87.93 ±0.00
Best W	W=0.6	W=0.2	W=0	W=0.1	W=0.6
TP weight type	Raw stress	Spearman rho	Sammon stress	Spearman rho	Raw stress
PCA, NN classifier	59.99 ±2.79	74.51 ±3.08	96.78 ±0.62	55.78 ±1.21	88.34 ±1.80
PCA, weighted NN	63.80 ±2,69	76.58 ±2.57	96.82 ±0.63	58,13 ±1.40	90.24 ±1.88
Best W	W=0.5	W=0	W=0.1	W=0	W=0.7
TP weight type	MRRE	Raw stress	MRRE	Sammon stress	Raw stress

variants, with the best combination of topology preservation indices and W values, are being included in this comparison. Once again it is worth to note that using proposed approach is in general beneficial for the performance of data mining procedures performed in reduced feature space. It is particularly recommended to use elimination of deformed datasets elements for nearest-neighbor classifier. It is a result of weak robustness of this classifier to noisy training sample members, which thanks to approach being introduced here, can be neglected.

5 Conclusion

This paper introduces novel scheme designed for high-dimensional tasks of intelligent data analysis performed in the reduced feature space. Our proposal is based on an observation that dimensionality reduction affects the topological structure of the datasets. It is consequently suggested here to use measures of topology preservation to improve data analysis procedures performed thereafter.

Introductory studies on method's performance conducted for selected datasets show that it offers promising efficiency. Further research in this area will involve studying the influence of dimensionality reduction procedure being employed in the first step on the beneficial effect of the proposed weighting scheme. The effect of the technique under consideration on the performance of other intelligent data analysis procedures, e.g. neural-network classifiers will be also investigated. Furthermore the guidelines for choosing proper weighting function, as well as setting a value of W for given dataset will be worked out. Additional experiments, for larger datasets (representing for example documents' content or gene expression data), are likewise planned to be performed.

Acknowledgements. The first author would like to express his gratitude to National Laboratory of Pattern Recognition, Chinese Academy of Science and Professor Bao-Gang Hu in particular for their support throughout the research.

The study is co-funded by the European Union from resources of the European Social Fund. Project PO KL "Information technologies: Research and their interdisciplinary applications", Agreement UDA-POKL.04.01.01-00-051/10-00.

This research was supported in part by PL-Grid Infrastructure.

References

1. Borg, I., Groenen, P.J.F.: Modern Multidimensional Scaling: Theory and Applications. Springer, Heidelberg (2010)
2. Everitt, B.S., Landau, S., Leese, M., Stahl, D.: Cluster Analysis. Wiley, New York (2011)
3. Furht, B., Escalante, A. (eds.): Handbook of Data Intensive Computing. Springer, Heidelberg (2011)
4. Karbauskaite, R., Dzemyda, G.: Topology Preservation Measures in the Visualization of Manifold-Type Multidimensional Data. Informatica 20, 235–254 (2009)
5. Kerdprasop, K., Kerdprasop, N., Sattayatham, P.: Weighted K-Means for Density-Biased Clustering. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2005. LNCS, vol. 3589, pp. 488–497. Springer, Heidelberg (2005)
6. Konig, A.: Interactive Visualization and Analysis of Hierarchical Neural Projections for Data Mining. IEEE Transactions on Neural Networks 11(3), 615–624 (2000)
7. Lee, J.A., Verleysen, M.: Nonlinear Dimensionality Reduction. Springer, New York (2007)
8. Łukasik, S., Kulczycki, P.: An Algorithm for Sample and Data Dimensionality Reduction Using Fast Simulated Annealing. In: Tang, J., King, I., Chen, L., Wang, J. (eds.) ADMA 2011, Part I. LNCS (LNAI), vol. 7120, pp. 152–161. Springer, Heidelberg (2011)

9. Łukasik, S., Kulczycki, P.: Using Topology Preservation Measures for High-Dimensional Data Analysis in a Reduced Feature Space. Technical Transactions 1-AC, 5–16 (2012) (in Polish)
10. van der Maaten, L.J.P., Postma, E.O., Herik, H.J.: Dimensionality Reduction: A Comparative Review. Tilburg University Technical Report, TiCC-TR 2009-005 (2009)
11. Parvin, H., Alizadeh, H., Minati, B.: A Modification on K-Nearest Neighbor Classifier. Global Journal of Computer Science and Technology 10, 37–41 (2010)
12. Sammon, J.W.: A Nonlinear Mapping for Data Structure Analysis. IEEE Transactions on Computers 18, 401–409 (1969)
13. Sammut, C., Webb, G.I. (eds.): Encyclopedia of Machine Learning. Springer, New York (2011)
14. Silva, V.D., Tenenbaum, J.B.: Global versus Local Methods in Nonlinear Dimensionality Reduction. In: Becker, S., Thrun, S., Obermayer, K. (eds.) Advances in Neural Information Processing Systems, vol. 15, pp. 705–712. MIT Press, Cambridge (2003)
15. UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml/`
16. Verleysen, M., François, D.: The Curse of Dimensionality in Data Mining and Time Series Prediction. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 758–770. Springer, Heidelberg (2005)

HRoBi – The Algorithm for Hierarchical Rough Biclustering

Marcin Michalak[1,2] and Magdalena Stawarz[1]

[1] Institute of Informatics, Silesian University of Technology
ul. Akademicka 16, 44-100 Gliwice, Poland
{Marcin.Michalak,Magdalena.Stawarz}@polsl.pl
[2] SOMAR S.A., ul. Karoliny 4, 40-186 Katowice, Poland
M.Michalak@somar.com.pl

Abstract. The article presents the new algorithm of biclustering, based on the rough biclustering foundations. Each rough bicluster is considered as the ordered pair of its lower and upper approximation. Notions of lower and upper bicluster approximation are derived from the Pawlak rough sets theory. Every considered discrete value in the data can be covered with more than one rough bicluster. The presented algorithm is hierarchical, so the number of biclusters can be controlled by the user.

Keywords: biclustering, rough sets, hierarchical clustering.

1 Introduction

Biclustering – named also as co-clustering, two-dimensional clustering or two-mode clustering – is the problem where we are grouping scalars from the two-dimensional matrix. This approach has been started in 70's in the last century [4] and is successfully applied in bioinformatics [3,9,10] or text mining [2].

The brief description of the goal of the biclustering is that we are looking for the subset(s) of row(s) and column(s) which intersection(s) points (point) the similar (or similar from the analysis point of view) values. It causes that with the biclustering we want to cover an interesting region with rectangles. Because the region that we want to be covered may not be rectangular one of the two extreme situation may happen: the bicluster is big but contains a lot of incorrect (from the analysis point of view) cells or the region is covered with a big amount of small (but correct) biclusters.

In the literature we can find the wide group of methods that allow us to simplify the description of the objects with the some kind of generalisation but with the cost of losing the accuracy. Two most popular branches are fuzzy sets [11] and rough sets [8] and their derivatives. In the first approach the membership function μ of the set is defined and it describes whether element x belongs to the set ($\mu(x) = 1$), does not belong to the set ($\mu(x) = 0$) or element belongs to the set „partially" ($\mu(x) \in (0, 1)$). In the second approach the set X is approximated with its lower and upper approximation which can be interpreted as follows: every element of the lower approximation $\underline{X}$ is surely the element of the X;

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 194–205, 2013.

every element that does not belong to the upper approximation $\overline{X}$ surely does not belong to the X.

In this paper the rough approach to the problem of biclustering data is presented. It is based on the previous works on exact biclustering [7] and rough representation of the bicluster [6]. It is designed for matrices with discrete values but its main idea is based on the biclustering of binary matrices. The algorithm generates rough biclusters in the hierarchical way and this explains its name: *HRoBi – **H**ierarchical **Ro**ugh **Bi**clustering.*

2 Related and Previous Works

Every matrix M contains rows and columns. There are several areas where there is no difference between considering rows as rows or rows as columns. It means, that the transposed matrix M' still contains the same information. After the transposition rows become columns and vice versa. In the paper [5] we have introduced the notion of feature and co-feature instead of rows and columns. If we consider features as rows than co-features automatically are considered as columns. If we consider features as columns than co-features are rows.

Single feature is denoted as f and the set of features as $\mathcal{F}$. Co-feature is denoted as f^* and all co-features as $\mathcal{F}^*$. $M\{f, f^*\}$ is the value in the matrix M and it may be considered as $M\{f, f^*\} = M(f, f^*)$ or $M\{f, f^*\} = M(f^*, f)$. The advantage of this generalisation is presented in the Fig. 1.

	c_1	c_2	c_3
r_1	1	2	3
r_2	4	5	6

Original table

	f_1^*	f_2^*	f_3^*
f_1	1	2	3
f_2	4	5	6

Rows as features

	f_1	f_2	f_3
f_1^*	1	2	3
f_2^*	4	5	6

Rows as co-features

Fig. 1. Illustration of features and co-features

If the original data is the matrix $M = [F, F^*]$ with features $\mathcal{F}$ and co-features $\mathcal{F}^*$ the bicluster $B = [\mathcal{G}, \mathcal{G}^*]$ is called the subset of M iff $\mathcal{G} \subseteq \mathcal{F}$ and $\mathcal{G}^* \subseteq \mathcal{F}^*$. This kind of inclusion will be denoted as binclusion (biclustering inclusion).

2.1 Exact Biclustering

As the exact biclustering – to be more precise the exact discrete biclustering – the problem of finding biclusters which elements have exactly the same value is denoted. In the previous works the two step solution of exact biclustering was developed [7].

The algorithm of exact biclustering (*eBi – **e**xact **Bi**clustering*) finds biclusters for every discrete value in the matrix separately. In its first step finds halfbiclusters that are subsets of features and co-features. Halfbicluster is generated for two features and is the largest – in the sense of inclusion – subset of co-features that makes it possible to not discern these two features from each other. Halfbicluster for two co-features is generated analogically. From every halfbicluster the corresponding bicluster is generated and both of them determine the bicluster.

The big number of overlapping exact biclusters is limited in the second step of the algorithm. The „from coverage" filtering – similar as in the case of decision rules filtering – is applied: in every step the ranking of bicluster coverage is calculated and the bicluster with the biggest coverage is moved to the result set. Then the ranking is recalculated and only the coverage of non-covered elements is taken into consideration. The filtration stops when all elements with the specified discrete value from the matrix are covered by the result set.

2.2 Rough Biclustering Foundations

In the paper [6] the notion of rough bicluster was defined for the first time. Starting from the $\beta-$relation $\beta \subseteq (2^{\mathcal{F}} \times 2^{\mathcal{F}^*})^2$ that joins subsets of features with subsets of co-features, which intersections in the matrix M have the same value v, the $\beta-$description classes were defined. Their meaning is analogous to the notion of equivalence class because $\beta-$description class is defined as an ordered pair of subsets of $\mathcal{F}$ and $\mathcal{F}^*$ as follows:

$$[(f, f^*)]_{\beta^v} = (F, F^*), \quad F \subseteq \mathcal{F}, F^* \subseteq \mathcal{F}^*$$

iff:

- $(f, f^*)\beta^v(F \times F^*)$
- $\forall_{e \notin F} \forall_{e^* \notin F^*} \ \neg\,[(f, f^*)\beta^v(F \cup \{e\}) \times (F^* \cup \{e^*\})]$

This means that the $\beta-$description class is the largest (in the sense of inclusion) subset of features and co-features which Cartesian product gives cells that are all in the β^v relation with the given one cell.

Because cells may have more than one $\beta-$description class we define the set of all $\beta-$description classes from matrix M which is called the dictionary: $\mathcal{D}_M^v$.

Let us consider the following relation $\mathcal{R}(\mathcal{D}_M^v) \subseteq \mathcal{D}_M^v \times \mathcal{D}_M^v$. Two $\beta-$description classes $d_1 = (F_1, F_1^*), d_2 = (F_2, F_2^*)$ are in the relation $\mathcal{R}(\mathcal{D}_M^v)$ when at least one of the following conditions is satisfied:

i) $F_1 \cap F_2 \neq \emptyset \wedge F_1^* \cap F_2^* \neq \emptyset$
ii) $\exists_{d_3 \in \mathcal{D}_M} \ d_1 \mathcal{R}(\mathcal{D}_M^v) d_3 \wedge d_3 \mathcal{R}(\mathcal{D}_M^v) d_2$

This relation is the equivalence relation. The partition of $\mathcal{R}$ introduced by this relation will be denoted as: $\Pi_M^v = \{\pi_1, \pi_2, \cdots, \pi_p\}$. This means that π_i is the set of $\beta-$description classes. For every π_i two sets may be defined, connected with predecessors and successors of pairs that are elements of π_i.

$$p(\pi_i) = \{F \subseteq \mathcal{F} : \exists_{F^* \subseteq \mathcal{F}^*} \ (F, F^*) \in \pi_i\}$$

$$s(\pi_i) = \{F^* \subseteq \mathcal{F}^* : \exists_{F \subseteq \mathcal{F}} \ (F, F^*) \in \pi_i\}$$

Every π_i generates the one rough bicluster b_i in the following way:

i) lower approximation: $\underline{b_i} = (\bigcap p(\pi_i), \bigcap s(\pi_i))$
ii) upper approximation: $\overline{b_i} = (\bigcup p(\pi_i), \bigcup s(\pi_i))$

More details can be found in [6]. The results of *eBi* algorithm are considered in this paper as the $\mathcal{D}_M$ and as the initial solution for their generalisation.

2.3 Biclusters Evaluations

When the matrix M has only values from the set $\{0, 1\}$ we assume that we want to find biclusters of ones excluding as much zeros as it is possible. In this approach biclusters may be described with the following measures:

- $w(M)$ - weight of M: number of ones in the matrix or bicluster,
- $\overline{\overline{\mathcal{B}}}$ - bicluster area: multiplication of numbers of features and co-features (if $\mathcal{B} = [\mathcal{G}, \mathcal{G}^*]$ then $\overline{\overline{\mathcal{B}}} = |\mathcal{G}| \cdot |\mathcal{G}^*|$,
- $acc(\mathcal{B}) = w(\mathcal{B})/\overline{\overline{\mathcal{B}}}$ bicluster accuracy: the ratio of bicluster weight and its area,
- $cov(\mathcal{B}) = w(\mathcal{B})/w(M)$ bicluster coverage: the ratio of bicluster weight and the whole matrix M weight.

As the rough bicluster is defined with two approximation biclusters (lower and upper) it may seem natural to define measures such as upper and lower accuracy and upper and lower coverage.

To assure the utility of the notion of lower approximation we demand that it is the exact bicluster - the one with accuracy equal to one. This means that for all rough biclusters their lower accuracy is one ($acc(\underline{\mathcal{B}}) = 1$) and for all biclusters the following relation is also true:

$$acc(\underline{\mathcal{B}}) \geq acc(\overline{\mathcal{B}})$$

When we consider lower and upper bicluster approximation the weight of upper approximation is greater or equal to the weight of the lower bicluster approximation.

$$w(\underline{\mathcal{B}}) \leq w(\overline{\mathcal{B}})$$

Dividing it with the weight of the interesting matrix M we obtain the following relation between coverages of bicluster lower and upper approximations:

$$cov(\underline{\mathcal{B}}) \leq cov(\overline{\mathcal{B}})$$

The exact bicluster is the one which accuracy is 1. The non-rough bicluster is the one which border $b(\mathcal{B}) = \overline{\mathcal{B}} \setminus \underline{\mathcal{B}} = \emptyset$. The full-rough bicluster is the one which $\underline{\mathcal{B}} = \emptyset$. The roughness is the measure that says how rough is the bicluster and is the fraction of the bicluster border and upper approximation capacities.

$$\rho(\mathcal{B}) = |b(\mathcal{B})|/|\overline{\mathcal{B}}| = 1 - |\underline{\mathcal{B}}|/|\overline{\mathcal{B}}| = 1 - w(\underline{\mathcal{B}})/|\overline{\mathcal{B}}|$$

It is possible to define the rough bicluster which lower and upper approximations are exact biclusters. It causes that its roughness is greater than zero, but from the utilitarian point of view we should treat it as the exact bicluster which lower and upper approximation are equal to the initial upper approximation.

Now, based on the fact that $w(\overline{\mathcal{B}}) \leq |\overline{\mathcal{B}}|$ we may say that:

$$w(\underline{\mathcal{B}}) \leq w(\overline{\mathcal{B}}) \leq |\overline{\mathcal{B}}|$$

Dividing all sides of inequalities by the size of upper approximation we obtain:

$$w(\underline{\mathcal{B}})/|\overline{\mathcal{B}}| \leq w(\overline{\mathcal{B}})/|\overline{\mathcal{B}}| \leq 1$$

We may notice that the inequality above may be written with the previously mentioned bicluster measures:

$$1 - \rho(\mathcal{B}) \leq acc(\mathcal{B}) \leq 1$$

This means that increase of inroughness brings the increase of accuracy and from the other hand the decrease of accuracy increases the roughness.

2.4 Rough Sum of Biclusters

For rough biclusters, which lower and upper approximations are an ordered pair of features and co-features, their rough sum is a rough bicluster which lower approximation is the pair of features from the both of lower approximations and co-features analogously and the upper approximation as the pair of sum of arguments features and a sum of arguments co-features:

$$\underline{\mathcal{A}} = [\{l_{fa_1}, l_{fa_2}, \ldots, l_{fa_k}\}, \{l_{ca_1}, l_{ca_2}, \ldots, l_{ca_m}\}]$$
$$\overline{\mathcal{A}} = [\{u_{fa_1}, u_{fa_2}, \ldots, u_{fa_n}\}, \{u_{ca_1}, u_{ca_2}, \ldots, u_{ca_o}\}]$$
$$\underline{\mathcal{B}} = [\{l_{fb_1}, l_{fb_2}, \ldots, l_{fb_i}\}, \{l_{cb_1}, l_{cb_2}, \ldots, l_{ca_j}\}]$$
$$\overline{\mathcal{B}} = [\{u_{fb_1}, u_{fb_2}, \ldots, u_{fb_x}\}, \{u_{cb_1}, u_{cb_2}, \ldots, u_{cb_v}\}]$$

$$\underline{\mathcal{A} \cup \mathcal{B}} = [\{l_{fa_1}, l_{fa_2}, \ldots, l_{fa_k}\} \cap \{l_{fb_1}, l_{fb_2}, \ldots, l_{fb_i}\},$$
$$\{l_{ca_1}, l_{ca_2}, \ldots, l_{ca_m}\} \cap \{l_{cb_1}, l_{cb_2}, \ldots, l_{ca_j}\}]$$
$$\overline{\mathcal{A} \cup \mathcal{B}} = [\{u_{fa_1}, u_{fa_2}, \ldots, u_{fa_n}\} \cup \{u_{fb_1}, u_{fb_2}, \ldots, u_{fb_x}\},$$
$$\{u_{ca_1}, u_{ca_2}, \ldots, u_{ca_o}\} \cup \{u_{cb_1}, u_{cb_2}, \ldots, u_{cb_v}\}]$$

We demand that lower approximation of the bicluster should always be exact. The operation of summing biclusters fulfils this criterion.

3 The *HRoBi* Algorithm

3.1 Lower and Upper Approximation Matrices

Let there is given a set of exact biclusters $\mathcal{EB} = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, ..., \mathcal{B}_e\}$ and $e = |\mathcal{EB}|$. Then the following matrix $LAM_{(e \times e)}$ (Lower Approximations Matrix) may be defined as follows: let us consider each bicluster from EB as the rough bicluster which lower and upper approximations are equal: $\mathcal{B}_i = [\underline{\mathcal{B}_i}, \overline{\mathcal{B}_i}]$ where $\underline{\mathcal{B}_i}$ is the lower approximation and $\overline{\mathcal{B}_i}$) is the upper approximation. Each cell in LAM is the exact bicluster that is the bicluster intersection of two rough biclusters lower

approximations: $LAM(i,j) = \underline{\mathcal{B}_i} \cap \underline{\mathcal{B}_j}$. In the same way the Upper Approximations Matrix may be defined: $UAM(i,j) = \overline{\mathcal{B}_i} \cup \overline{\mathcal{B}_j}$.

The merge of two rough biclusters $\mathcal{B}_i$ and $\mathcal{B}_j$ is the rough bicluster which approximations are taken from the LAM and UAM:

$$\mathcal{B}_{ij} = \mathcal{B}_i \circ \mathcal{B}_j = [LAM(i,j), UAM(i,j)]$$

In other words:

$$\underline{\mathcal{B}_{ij}} = LAM(i,j), \quad \overline{\mathcal{B}_{ij}} = UAM(i,j)$$

Only biclusters $\mathcal{B}_i$ and $\mathcal{B}_j$ which intersection in LAM is different from empty set should be considered to be merged.

3.2 Hierarchical Rough Biclustering

The algorithm of hierarchical rough biclustering is based on rough sum of biclusters. Starting from the set of exact biclusters – that can be easily treated as the rough biclusters which upper and lower approximations are equal – the ranking of rough biclusters is built: on the top of the ranking is the rough bicluster with the smallest roughness.

The new rough bicluster substitutes two origin one in the set and the procedure follows. The algorithm stops when the one from two conditions is fulfilled: there is only one rough bicluster in the set (all biclusters were joined) and all existing rough biclusters have empty intersection in pairs of their lower approximations.

In the second case it is possible to continue the procedure of joining biclusters (with the generalisation of the rough sum of biclusters) but it will cause that the lower approximation of biclusters generated in that way will be always empty. In our opinion it is more important to assure a minimal interpretability of the rough bicluster.

Avoiding joining biclusters which lower approximations intersection is the empty set gives also the possibility to represent the real structure of the data in the matrix: if there are regions that are not continuous they should not be represent as the one bicluster.

4 Case Study

Let us analyse step by step the idea of hierarchical rough biclustering. The starting point is the set of exact biclusters generated by the *eBi*. For the given small matrix (shown on the Table 1) *eBi* generated four biclusters (Table 2). Biclusters are marked with black cells.

In the first iteration biclusters B_2 and B_4 are joined as the rough sum of rough biclusters. They generate the $B_{2,4}$ bicluster (Table 3). The range of the lower approximation is marked with underlined values and the range of the upper approximation is marked with overlined values.

In the following iteration the bicluster B_1 and bicluster $B_{2,4}$ are joined (Table 4). Now we see that the algorithm stops when the set of biclusters contains one rough and one non-rough bicluster. It is also possible that the final set will contain several rough biclusters which would not be joined because each pair of them will have an empty intersection of their lower approximations.

Table 1. Case study matrix M

	f_1^*	f_2^*	f_3^*	f_4^*	f_5^*
f_1	0	0	0	0	0
f_2	0	1	1	1	0
f_3	0	0	1	1	0
f_4	1	1	1	1	0
f_5	0	0	0	0	1

Table 2. Four exact biclusters of matrix M (from left to right: B_1, B_2, B_3, B_4)

B_1:

	f_1^*	f_2^*	f_3^*	f_4^*	f_5^*
f_1	0	0	0	0	0
f_2	0	1	1	1	0
f_3	0	0	1	1	0
f_4	1	1	1	1	0
f_5	0	0	0	0	1

B_2:

	f_1^*	f_2^*	f_3^*	f_4^*	f_5^*
f_1	0	0	0	0	0
f_2	0	1	1	1	0
f_3	0	0	1	1	0
f_4	1	1	1	1	0
f_5	0	0	0	0	1

B_3:

	f_1^*	f_2^*	f_3^*	f_4^*	f_5^*
f_1	0	0	0	0	0
f_2	0	1	1	1	0
f_3	0	0	1	1	0
f_4	1	1	1	1	0
f_5	0	0	0	0	1

B_4:

	f_1^*	f_2^*	f_3^*	f_4^*	f_5^*
f_1	0	0	0	0	0
f_2	0	1	1	1	0
f_3	0	0	1	1	0
f_4	1	1	1	1	0
f_5	0	0	0	0	1

Table 3. Results after the first iteration: two exact and one rough bicluster of matrix M (from left to right: $B_1, B_{2,4}, B_3$)

	f_1^*	f_2^*	f_3^*	f_4^*	f_5^*
f_1	0	0	0	0	0
f_2	0	1	1	1	0
f_3	0	0	1	1	0
f_4	1	1	1	1	0
f_5	0	0	0	0	1

	f_1^*	f_2^*	f_3^*	f_4^*	f_5^*
f_1	0	0	0	0	0
f_2	$\overline{0}$	$\overline{1}$	$\overline{1}$	$\overline{1}$	0
f_3	$\overline{0}$	$\overline{0}$	$\overline{1}$	$\overline{1}$	0
f_4	$\overline{1}$	$\overline{1}$	$\overline{\underline{1}}$	$\overline{\underline{1}}$	0
f_5	0	0	0	0	1

	f_1^*	f_2^*	f_3^*	f_4^*	f_5^*
f_1	0	0	0	0	0
f_2	0	1	1	1	0
f_3	0	0	1	1	0
f_4	1	1	1	1	0
f_5	0	0	0	0	1

5 Experiments

To compare the rough and exact biclustering results the experiments on synthetic data were performed. The matrix is 100x100 and contains three discrete values: (#0 – 1415 cells), grey (#77 – 1327 cells, three areas) and light grey (#237 – 2148 cells, banana shape). The white region is considered as the background and is not the object of interest. Level areas do not have sharp edges and some regions of both grey colors overlap. The matrix is presented on the Fig. 2.

Table 4. Results after the second (the last) iteration: one rough ($B_{1,2,4}$) and one exact (B_3) bicluster of matrix M

	f_1^*	f_2^*	f_3^*	f_4^*	f_5^*
f_1	0	0	0	0	0
f_2	$\overline{0}$	$\overline{1}$	$\overline{1}$	$\overline{1}$	0
f_3	$\overline{0}$	$\overline{0}$	$\overline{1}$	$\overline{1}$	0
f_4	$\overline{1}$	$\overline{1}$	$\underline{1}$	$\underline{1}$	0
f_5	0	0	0	0	1

	f_1^*	f_2^*	f_3^*	f_4^*	f_5^*
f_1	0	0	0	0	0
f_2	0	1	1	1	0
f_3	0	0	1	1	0
f_4	1	1	1	1	0
f_5	0	0	0	0	1

Fig. 2. Synthetic data

Three algorithms of non-rough biclustering were compared with *HRoBi*. Order Preserving Submatrix Algorithm *OPSM* [1] defines bicluster as an order preserving submatrix of dataset, therefore rows and columns, which belong to biclusters can be sorted in ascending order. Based on own probabilistic model for simulation of real data, *OPSM* can find large and statistically significant biclusters, which satisfy its rigorous requirements. To solve the NP-hard problem of finding biclusters, heuristic approach was applied.

Bicluster in Cheng and Church's Algorithm (*CC*) [3] is described as subset of rows and subset of columns for which Mean Squared Residue Score (*MSRS*) is under threshold, set by user. In order to overcome the problem of finding the largest biclusters, two phase strategy has been used. The main concept is to repeat two steps, while assumed condition of end is not satisfied. The first step consists of deleting rows and columns from dataset, and the second step follows by insertion of rows and columns deleted in the previous step into data. Steps are repeated until the *MSRS* is below defined level. Algorithm selects biclusters applying the greedy search strategies. As the predefined number of biclusters the following ones were taken into consideration: 10, 40, 70, 100, 130, 160, 190, 220, 250, 280 and 310.

The only one modification of *eBi* in respect to the description in [7] is that the result set is obtained with the double filtration. In the first step exact biclusters

are filtered after being generated in the f and f^* mode separately and then the sum of these sets is filtered again.

Tables 5, 6 and 7 show results of biclustering with the *OPSM*, *CC* and *eBi* algorithm respectively.

Table 5. Statistics of biclustering results from CC algorithm for three discrete values (0, 77, 237) for different predefined numbers of biclusters (Number of Biclusters – the CC algorithm parameter.).

Number of Biclusters	v-value	total number	avg. acc.	avg. cov.	sum. cov.	total cov.
10	0	0	0.00	0.00	0.00	0.00
40	0	9	0.25	0.01	0.13	0.10
70	0	20	0.29	0.02	0.41	0.35
100	0	23	0.27	0.02	0.42	0.36
130	0	25	0.27	0.02	0.46	0.38
160	0	27	0.26	0.02	0.47	0.38
190	0	35	0.24	0.01	0.50	0.40
220	0	46	0.24	0.01	0.58	0.45
250	0	52	0.23	0.01	0.64	0.46
280	0	57	0.23	0.01	0.69	0.47
310	0	63	0.23	0.01	0.74	0.47
10	77	3	0.38	0.06	0.17	0.17
40	77	22	0.34	0.02	0.50	0.46
70	77	37	0.3	0.02	0.72	0.59
100	77	50	0.3	0.02	0.8	0.61
130	77	68	0.27	0.01	0.97	0.62
160	77	82	0.29	0.01	1.07	0.63
190	77	98	0.27	0.01	1.17	0.65
220	77	113	0.26	0.01	1.24	0.66
250	77	129	0.26	0.01	1.36	0.67
280	77	151	0.26	0.01	1.49	0.67
310	77	169	0.26	0.01	1.66	0.68
10	237	9	0.32	0.06	0.55	0.55
40	237	37	0.31	0.02	0.79	0.73
70	237	60	0.32	0.02	0.94	0.79
100	237	86	0.33	0.01	1.07	0.82
130	237	110	0.34	0.01	1.25	0.83
160	237	133	0.34	0.01	1.36	0.84
190	237	154	0.34	0.01	1.52	0.85
220	237	174	0.34	0.01	1.63	0.85
250	237	198	0.34	0.01	1.77	0.87
280	237	224	0.34	0.01	1.92	0.87
310	237	249	0.34	0.01	2.09	0.88

Table 6. Statistics of biclustering results from OPSM algorithm

$v-$ value	total number	avg. acc.	avg. cov.	sum. cov.	total cov.
#0	27	0.15	0.17	3.41	0.43
#77	33	0.12	0.09	3.07	0.48
#237	40	0.21	0.07	2.76	0.28

Table 7. Statistics of biclustering results from *eBi* algorithm after postprocessing

$v-$value	total number	avg cov	sum. cov
#0	77	0.09	7.14
#77	121	0.08	9.47
#237	144	0.07	10.07

As the *avg. acc.* the average accuracy of biclusters is denoted. The *avg. cov.* column means the average coverage of biclusters. The *sum. cov.* column means

the sum of coverage of all biclusters so the value greater than one points that the total area of biclusters is bigger than the area of the considered discrete value. The *total cov.* column says which part of cells with the considered discrete value is covered by the set of obtained biclusters. Results of *eBi* do not contain columns *avg. acc.* and *total cov.* It is due to the fact that each bicluster generated by the *eBi* algorithm is exact (its accuracy is equal to one) and all considered discrete values are covered (proofs of these properties can be found in [7]).

Results of rough biclustering are shown in the Table 8. We may see that the algorithm stopped with more than one final bicluster. That means that for every considered discrete value each pair of rough biclusters from the result set has an empty intersection of their lower approximations.

Table 8. Statistics of biclustering results from *HRoBi* algorithm

$v-$value	total number	avg. acc	avg. cov	sum. cov.
#0	10	0.77	0.27	2.65
#77	13	0.77	0.20	2.63
#237	12	0.72	0.29	3.50

If we compare the *HRoBi* results to *eBi* results we observe the big decrease of the number of biclusters and the significant decrease of the level of biclusters overlapping (measured as the sum of all biclusters coverage) together with the increase of the average biclusters coverage. The natural consequence of summing biclusters in the rough way is the decrease of bicluster accuracy.

Because *HRoBi* is the hierarchical algorithm and its results can be evaluated after every iteration separately, on the Fig. 3 some statistics describing every partial results are shown. The axis x is the number of performed iterations. After each iteration the following statistics are calculated for the remaining rough biclusters — minimal, maximal, mean and the standard deviation of:

- accuracy of the upper approximation (au),
- coverage of the upper approximation (cu),
- coverage of the lower approximation (cl),
- roughness of the rough bicluster (ρ).

Charts start from 0 because they represent statistics for the initial set of biclusters (obtained from *eBi*). In the following iterations we can observe a decrease of the accuracy of the upper approximation and increase of the roughness, which is the natural consequence of combining exact biclusters into one. However, thanks to the assumption, that *HRoBi* choose always two biclusters, which joined together give the smallest roughness, the algorithm ensures minimum loss of the accuracy, when the coverage of upper approximation is increasing. The coverage of lower approximation is decreasing, but at the end is never equal to zero, what provides that rough biclusters found by *HRoBi* will always cover at least one analysed value. Finally, the whole matrix (all discrete values) from our experiment dataset can be described with only 35 rough biclusters. Their mean coverage is between 0.20-0.29 and they loss only 23-28 % from their accuracy (Table 8) when compared to *eBi* results (Table 7).

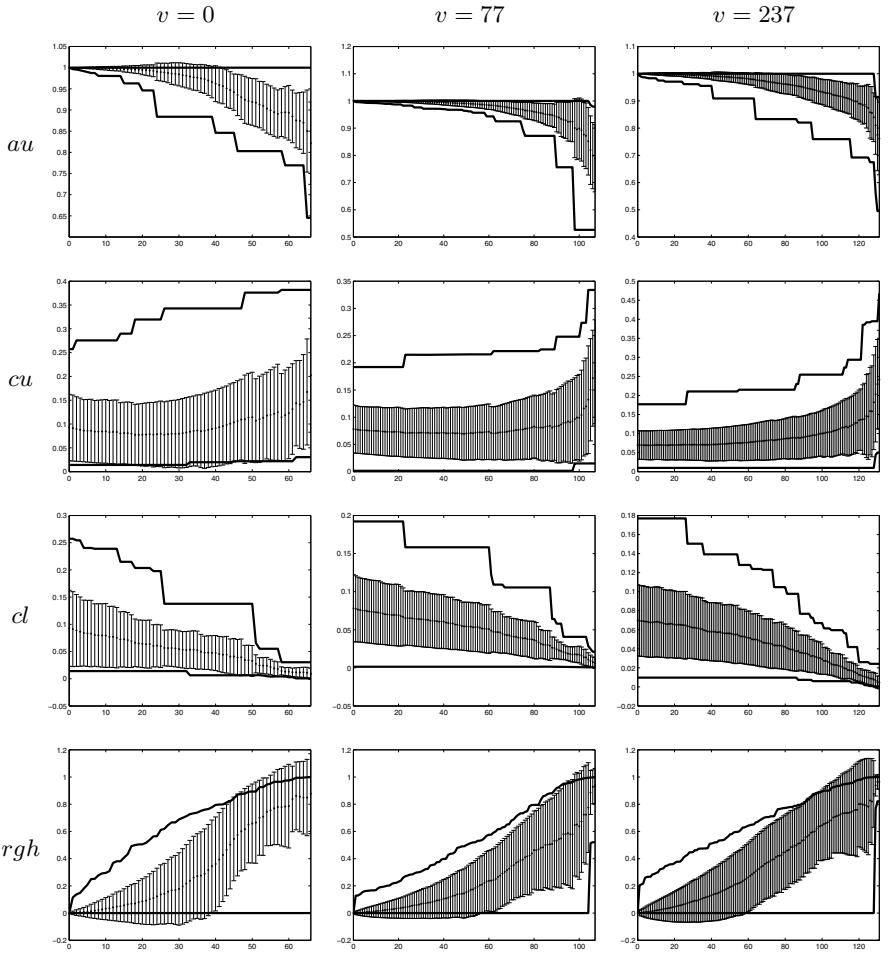


Fig. 3. Averaged statistics of *HRoBi* results: au – accuracy of the upper approximation, cu – coverage of the upper approximation, cl – coverage of the lower approximation, ρ – roughness of the rough bicluster. Solid lines are maximal and minimal values of the statistics, dotted lines are mean values and bars are the standard deviations of mean values.

6 Conclusions

In this article the new algorithm of the hierarchical biclustering based on the rough sets theory has been presented. Based on the exact biclusters found by the *eBi*, the *HRoBi* algorithm generates rough biclusters. In each iteration of *HRoBi* two biclusters are merged if the intersection of their lower approximations is not empty and the value of roughness is the lowest. The most important feature of this method is the ability of finding biclusters with less accuracy but bigger

coverage, what is in the opposition to the *eBi* algorithm, which generates exact but small biclusters. Other algorithms (*CC* and *OPSM*), which were compared with the *HRoBi* do not achieve such a satisfactory results - small number of big biclusters, covering all considered discrete values, with still quite high accuracy. The *HRoBi* algorithm seems to be very useful for the purpose of big datasets analysis due to two capabilities. The first one is the ability to find rough biclusters which coverage and accuracy is above or under some threshold defined by user. The second is a substantial simplification in the representation of distribution of discrete values in the matrix using biclusters. Due to the high computational complexity of the algorithm our next goal is the parallelisation which can be applied in several aspects: biclusters for every discrete value are computed in the same time, building the ranking of biclusters to be summed or building LAM and UAM matrices.

Acknowledgements. This work was supported by the European Union from the European Social Fund (grant agreement number: UDA-POKL.04.01.01-00-106/09).

References

1. Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering Local Structure in Gene Expression Data: The Order-Preserving Sub-Matrix Problem. J. of Comput. Biol. 10(3-4), 373–384 (2003)
2. Chang, F.C., Huang, H.C.: A refactoring method for cache-efficient swarm intelligence algorithms. Information Sciences 192, 39–49 (2012)
3. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proc. of the 8th Int. Conf. on Intell. Syst. for Mol. Biol., pp. 93–103 (2000)
4. Hartigan, J.A.: Direct Clustering of a Data Matrix. J. Am. Stat. Assoc. 67(337), 123–129 (1972)
5. Michalak, M., Stawarz, M.: Generating and Postprocessing of Biclusters from Discrete Value Matrices. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part I. LNCS, vol. 6922, pp. 103–112. Springer, Heidelberg (2011)
6. Michalak, M.: Foundations of Rough Biclustering. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 144–151. Springer, Heidelberg (2012)
7. Stawarz, M., Michalak, M.: eBi – The Algorithm for Exact Biclustering. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 327–334. Springer, Heidelberg (2012)
8. Pawlak, Z.: Rough Sets. J. of Comput. and Inf. Sci. 5(11), 341–356 (1982)
9. Pensa, R., Boulicaut, J.F.: Constrained Co-clustering of Gene Expression Data, Proc. SIAM Int. Conf. on Data Min., SDM 2008, pp. 25–36 (2008)
10. Yang, E., Foteinou, P.T., King, K.R., Yarmush, M.L., Androulakis, I.P.: A Novel Non-overlapping biclustering Algorithm for Network Generation Using Living Cell Array Data. Bioinforma 17(23), 2306–2313 (2007)
11. Zadeh, L.H.: Fuzzy Sets. Information and Control 8(3), 338–353 (1965)

Proximity Measures and Results Validation in Biclustering – A Survey

Patryk Orzechowski

AGH University of Science and Technology,
Department of Automatics and Biomedical Engineering
30-059 Cracow, Mickiewicza Av. 30, Poland
patrick@agh.edu.pl
http://home.agh.edu.pl/~patrick

Abstract. The concept of biclustering evolved from traditional clustering techniques, which have proved to be inadequate for discovering local patterns in gene microarrays, in particular with shifting and scaling patterns. In this work we compare similarity measures applied in different biclustering algorithms and review validation methodologies described in literature. To our best knowledge, this is the first in-depth comparative analysis of proximity measures and validation techniques for biclustering. Current trends in design of similarity measures as well as a rich collection of state-of-the-art benchmark datasets are presented, supporting algorithm designers in classification of comparison and quality assessment criteria of emerging biclustering algorithms.

Keywords: biclustering, co-clustering, shifting and scaling patterns, pattern similarity, proximity measures, results validation, microarray gene expression data, state-of-the-art, survey.

1 Introduction

Gene expression data (or simply: *expression data*) is usually organized in form of a matrix with rows corresponding to different genes and columns to conditions. Data samples come from individual organs, tissues (healthy or affected by some disease) or organisms exposed to various environmental conditions or a single condition over certain amounts of time. Genes that react similarily to certain conditions are supposed to have corresponding functionalities and be involved in similar biological processes [38].

The concept of biclustering (formulated by Hartigan [24] and applied primarily to gene expression data by Cheng and Church [12]) emerged as clustering algorithms were inadequate to detect similar expressions of genes exhibited to certain set of (not all) conditions. Clustering managed to detect similar patterns only on a global scale and grouped either all genes by analysing they response to all conditions, or conversely grouped all conditions by taking into account whole expression profiles of the genes. Another problem was clustering complexity, as clustering process had to be applied separately to columns and rows of the data

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 206–217, 2013.

matrix. Afterwards, like in hierarchical clustering [15], typically all combinations of rows/columns had to be considered so that a subset of rows and columns satisfying criteria could be determined. Keeping in mind the fact that for a n element set we have 2^n combinations in total, biclustering approaches that approached both dimensions simultaneously were a natural consequence.

In this article we summarize developments in biclustering and analyse current trends in evolution of similarity measures that determine algorithm architecture.

2 Definitions

Given a n x m data matrix $A = \{a_{ij}\}$ with values obtained by exposing n objects $X = \{x_1, ..., x_n\}$ to m conditions $Y = \{y_1, ..., y_m\}$, the response of the *i-th* gene to the *j-th* condition can be described as $a_{ij} = (x_i, y_j)$, where $i \leq n$ and $j \leq m$.

Formally, biclustering problem may be formulated as follows. Given any n x m data matrix $A = \{a_{ij}\}$ with rows $X = \{x_{1,}..., x_n\}$ and columns $Y = \{y_{1,}..., y_m\}$, its values could be presented as $a_{ij} = (x_i, y_j)$. Subset of p rows $I = \{i_{1,...,}i_p\}$, where $I \subseteq X$ and $p \leq n$ and subset of q columns $J = \{j_{1,...,}j_q\}$, where $J \subseteq Y$ and $q \leq m$ is called a *bicluster* $B = (I, J) = \{a_{ij} \in A : i \in I, j \in J\}$ when its rows I are as similar as possible to each other across its columns J and vice-versa. *Biclustering* is task of identifying a series of *biclusters* $B_k = (I_k, J_k)$, such that each B_k would meet a specified homogeneity (or similarity) criterion [38].

The second definition of bicluster [1] defines bicluster as a triplet $B(I, J, h)$, in which $I \subseteq X$, $J \subseteq Y$ and $h : I \times J \to R$ is the level function of the bicluster, such that $\forall_{(x_i,y_j)\in I\times J} : h(x_i, y_j) = a_{ij}$. Basing on this definition, biclustering aims at identification of triplets.

Biclustering formulation varies across authors, though, with some wanting to obtain motifs [40], plaid models [34], perfect δ -biclusters [38] etc.

One of the main challenges of biclustering is to find biclusters with shifting and scaling patterns. A group of genes show a *shifting pattern* (when the values of genes vary with the addition of an additive constant β_i) or *scaling pattern* (when the values vary with multiplication by a multiplicative constant α_i).

Formally, a bicluster $B = \{a_{ij}\}$ has a shifting or scaling pattern when it follows the expressions (1) or (2), respectively:

$$a_{ij} = \pi_j + \beta_i \tag{1}$$

$$a_{ij} = \pi_j \times \alpha_i \tag{2}$$

where the values β_i and α_i are fixed for all genes and π_j is a fixed value for every condition. With random noise ε_{ij} included, elements of bicluster that follow both expressions (1) and (2) could be defined in general form as (3):

$$a_{ij} = \pi_j \times \alpha_i + \beta_i + \varepsilon_{ij} \tag{3}$$

Shifting and scaling patterns are of much interest, as genes may respond to conditions similarly, even though they may have started with different initial conditions or the level of their response may differ in strength [1]. Plaid model [34] and Bayesian Biclustering (BBC) [22] extend this concept.

3 Proximity Measures

A perfect biclustering algorithm needs to be in line with the following rules [33]:

- handle high dimensional data
- be insensitive to outliers, noise and order of data input
- have low complexity (time and space)
- require few input parameters
- incorporate meta-data knowledge
- produce biologically interpretable results

Many different measures have been tested for clustering or biclustering. Some of the commonly used measures have been categorized in this chapter.

3.1 Distance-Based Measures

Distance functions, including the so-called Minkowski measures (Euclidean, Manhattan, Chebyshev) are useful when extracting exact matches between two objects [33]. They are easily computable and yield global similarities between two vectors. Unfortunately, they are also very sensitive to noise and outliers [5][33]. Distance measures are excessively documented in [13]. More sophisticated conceptual measures need to be applied in order to overcome distance-based measure disadvantages [33].

3.2 Qualitative Measures

Qualitative measures often assume that positive and negative values in microarray data carry equal amount of information. Usually, these are numbers of ups/down/no changes or (in binary case) number of positive/negative values for conditions [5][13]. The most popular coefficients take into account the numbers of positive elements in both objects (denoted as a), negative elements in both objects (denoted as d), number of positives in $j-th$ objects and negatives in $k-th$ object (denoted as b) or conversely (denoted by c). Qualitative measures are also classified as part of non-correlation measures [13].

The most widely applied qualitative measures are the so-called simple coefficient measure, *Jaccard coefficient* and *Sorenson coefficient*. Simple matching coefficient is defined as (4):

$$C_{jk} = \frac{a+d}{a+b+c+d}, \quad 0 \leq C_{jk} \leq 1 \tag{4}$$

Jaccard coefficient is defined for two objects j and k as follows (5):

$$J_{jk} = \frac{a}{a+b+c}, \quad 0 \leq J_{jk} \leq 1 \tag{5}$$

Sorenson coefficient between objects j and k is defined as (6):

$$S_{jk} = \frac{2a}{2a+b+c}, \quad 0 \leq S_{jk} \leq 1 \tag{6}$$

Maximum similarity is achieved when coefficients are equal to one. All coefficients fall within the range of 0 to 1. It is worth noticing how the last two coefficients are sensitive to the so-called direction of coding (switching all 0's and 1's would result in different coefficients values) [46].

The QUBIC [35] concept of sorting microarray values, taking only extreme values into account and later representing similarity as the number of element co-occurrences, extends qualitative measures. Such measure is used by ISA as well [28].

3.3 Non-correlation-Based Measures

The group of non-correlation measures contains algorithms which do not rely on counting elements of the set (nor their occurrences) and specify a formula for determining object similarity instead. One group of algorithms is dominant in this category – numerous approaches based on the measure proposed by Cheng-Church, namely Mean Square Residue [12]. Mean square residue H of bicluster (I, J) is defined by equation (7):

$$H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2 \tag{7}$$

where $a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}$ is a row mean, $a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}$ is a column mean and $a_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} a_{ij} = \frac{1}{|I|} \sum_{i \in I} a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{Ij}$ is the mean of submatrix (I, J) .

The second representative of non-correlation-based measures is HARP [57] that uses quality metric assessment and detects constant patterns only, which is not suitable for real data [7]. Many algorithms use Mean Square Residue straightforwardly or as a part of the determinant of similarity. These algorithms include Cheng-Church algorithm [12], its improvement – FLOC algorithm [56] and multiple recent publications, such as CPB [8], Particle Swarm Optimization [37], Simulated Annealing [9], greedy randomized adaptive search [14], localization and extraction with adaptive noise hiding LEB [17] or estimation of distribution algorithms [36].

It has been proven, however, that mean square residue manages shifting patterns (i.e. addition of a constant), but does not manage scaling patterns (i.e. multiplication by a constant might affect the score). This occurs when variance of gene values is high [1][5][47]. It suggests that the measure may have already become insufficient for biclustering purposes.

3.4 Correlation-Based Measures

There are two groups of measures using correlation: *parametric* (which estimate population parameters and assume bivariate normal distribution of data) and *non-parametric* (which allow less demanding assumptions and do not attempt to estimate population parameters) [11]. Correlation-based measures are scale-invariant, with distance between two objects usually calculated as $distance = 1 - correlation^2$ if sign is accurate [33].

Parametric Correlation-Based Measures. Two most commonly recognized parametric correlation measures are cosine similarity (also reckoned as distance measure, but classified in this category basing on its similarity to other representatives) and standard Pearson correlation. Cosine similarity between two vectors X and Y is defined as (8):

$$\cos(\theta) = \frac{X \cdot Y}{\|X\| \, \|Y\|} = \frac{\sum_{i=1}^{n} X_i \cdot Y_i}{\sqrt{\sum_{i=1}^{n} (X_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (Y_i)^2}} \tag{8}$$

Definition of traditional correlation includes means of both vectors (9):

$$\begin{aligned} cov(X,Y) = \sigma_{xy} = E((X - \overline{X})(Y - \overline{Y})) = \\ = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n} (X_i - \overline{X})^2} \cdot \sqrt{\sum_{i=1}^{n} (Y_i - \overline{Y})^2}} \end{aligned} \tag{9}$$

Pearson measure of correlation is defined as (10) [54]:

$$\rho = \frac{cov(X,Y)}{\sigma_x \sigma_y} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - (E(X))^2} \cdot \sqrt{E(X^2) - (E(X))^2}} \tag{10}$$

Parametric measures are said to be sensitive to outliers and noise [33][46] and to fail to capture true grouping [47] in opposition to non-parametric correlation based measures which base only on the ordinal position of elements. Some of recognized algorithms use modified Pearson correlation coefficient as the measure of similarity, BBC[22] and CPB [8] and Scatter Search [42] are examples.

Non-parametric Correlation Based Measures. Non-parametric correlation measures include i.a. Spearman's rank correlation and Kendall's τ. They are sometimes classified as distance measures [5].

Spearman's rank correlation coefficient (11), called also Spearman's rho coefficient, is considered to be equal to Pearson correlation between ranked variables. First, X and Y raw scores are ranked into x_i and y_i (taking into account average positions in the ascending order of the values), then Spearman's rank is calculated [41].

$$\rho = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2 \cdot \sum_{i=1}^{n} (y_i - \overline{y})^2}} \tag{11}$$

Some recently published similarity functions base on Spearman's rank correlation, for example Average Spearman's Rho (ASR) defined as (12):

$$ASR(I,J) = 2 \; max \left\{ \frac{\sum_{i \in I} \sum_{j \in I, j>i} \rho_{ij}}{|I|(|I|-1)}, \frac{\sum_{k \in J} \sum_{l \in J, l>k} \rho_{ij}}{|J|(|J|-1)} \right\} \tag{12}$$

where ρ_{ij} and ρ_{kl} are the Spearman's rank correlantions associated respectively with row indices i and j, and column indices k and l of a bicluster (I, J) [5].

Another approach basing on Spearman's rho is Average Correlation Value (ACV) defined as (13) [53]:

$$ACV(I,J) = max\left\{\frac{\sum_{i\in I}\sum_{j\in I}|\rho_{ij}| - |I|}{|I|(|I|-1)}, \frac{\sum_{k\in J}\sum_{l\in J}|\rho_{kl}| - |J|}{|J|(|J|-1)}\right\} \quad (13)$$

where ρ_{ij} and ρ_{kl} are defined analogically. Notice that $ASR(I,J) \in [-1;1]$ and $AVR(I,J) \in [0;1]$.

Second most recognizable non-parametric covariation-based measure is Kendall's τ defined as (14), which is said to be generally insensitive to outliers (i.e. outliers are detectable, but can change the value of correlation):

$$\tau = \frac{2\sum_{i<j} K_{ij}}{n(n-1)} \quad (14)$$

where

$$K_{ij} = \begin{cases} 1 & \text{when when } x_i \text{ and } y_i \text{ are concordant} \\ -1 & \text{when } x_i \text{ and } y_i \text{ are disconcordant} \end{cases}$$

Concordance between two samples is understood as agreement in order, i.e. two points $P = (x_i, y_j)$ and $Q = (x_j, y_j)$ are concordant if $(x_i - x_j)(y_i - y_j) > 0$.

Non-paramteric covariation-based measures are represented by FABIA [25] and PDNS[5]. Two more correlation measures are promising for biclustering and worth mentioning: *Mahalanobis distance* (scale invariant, with ellipsoidal shapes of clusters, instead of spherical as in Euclidean distance calculation) and *adaptive distance norm* from Gustafson-Kessel method of fuzzy clustering (with covariances estimated in eigenvalue calculations, unique distance measure applied to each cluster) [29][33].

4 Result Validation Methodology

Different problem formulations used in biclustering schemes impede general comparison of biclustering algorithms [45]. As algorithms are designed to work with specific constraints, they may perform better in a specific scenario and worse in others. Choice of correct initial parameters is also crucial to obtain satisfactory biclustering results [16].

The majority of biclustering algorithms refer to biological data and start with identification of locally co-expressed genes. Classification of samples or inference of regulatory mechanism are also areas of biclustering application [6][12][17][28][35][51].

The results obtained from biclustering algorithms may be validated basing on three index categories: *internal*, *external* and *relative* [23][30][29][31][45]. Internal indices verify if the recovered structure is appropriate for the data, external indices compare the recovered structure to the structure known *a priori*, while relative indices assess which of the recovered structures is better according to some quality measure.

4.1 Internal Indices

The number of biclusters, as well as sizes (minimum, maximum, average) of biclusters returned by an algorithm are first choice criteria to classify biclustering algorithms [17]. Such measures are useful for quantity assessment, but they do not involve the quality of each bicluster nor completeness of biclustering. Two clustering measures could be adapted to present (dis-)simmilarity of biclusters: *homogeneity* – the degree of similarity of elements in the same cluster and *separation* – the value determining the similarity of different clusters (i.e. how much biclusters overlap between each other) [20].

Homogeneity and Separation. *Homogeneity* represents average distance between objects in the same bicluster, while separation is defined as the weighted average similarity between objects from different biclusters [20]. *Homogeneity H* of an object u belonging to the bicluster X can be determined according to (15):

$$H = \frac{1}{N} \cdot \sum_{u \in N} f(g(u), g(X)) \tag{15}$$

where f is a similarity function and g is the expression level of an object.

Separation S for biclusters $X_1, ..., X_n$ is defined as (17) [49]:

$$S = \frac{1}{\sum_{i \neq j} |X_i||X_j|} \cdot \sum_{i \neq j} |X_i||X_j| f(g(X_i), g(X_j)) \tag{16}$$

Superior algorithms provide high homogeneity and low separation [49].

Other Measures. *Significance* adapted from clustering may be considered a third type of internal index. Basing on Monte Carlo method or probability assessment, significance assesses the likelihood of obtaining randomly a bicluster of given quality [30]. *Average silhouette width* may also be considered as a criterion, determining the distance of each object from other objects in a bicluster [20][10]. Several other measures for clustering algorithms are available in [31].

4.2 External Indices

Biological Datasets. The information used for biological validation is concerned to be external to data. Gene annotation databases, such as Gene Ontology (GO) [4] or KEGG [43][32], are common choice for performing validation of biclustering algorithms with respect to biological knowledge.

Most biclustering algorithms carry out experiments on one (or more) of the following datasets:

- yeast cell-cycle data set of *Saccharomyces cerevisiae*(selection of 2993 genes with 173 different stress conditions) [19][50]
- cancer dataset: 4026 genes of over 96 human tissue samples with 9 types of lymphoma and control [2][51]

- 12533 probes from 72 patients with different subtypes of leukemia [3][35]
- gene expression datasets provided by Broad Institute as "Cancer Program Data Sets", analyzed in [26][25]
- bio-synthetic pathways of *Arabidopsis thaliana* 734 genes under 69 experimental conditions [55]
- M3D database (4217 *Escherichia coli* genes under 264 conditions) [18][35]
- metabolic pathways [28], promoter motifs [27] etc.

Gene enrichment is measured with p-values, which specify the probability of finding the number of genes from a particular GO category (function, process and component) within each bicluster. In order to calculate probability p of finding at least k-genes from a category within a bicluster of size n, cumulative hyper-geometric distribution is used[52]:

$$p = 1 - \sum_{i=0}^{k-1} \frac{\binom{m}{i}\binom{g-m}{n-i}}{\binom{g}{n}} \tag{17}$$

Correspondence with GO category requires calculation of p-values for each functional category in each bicluster [37]. There exists some controversies as to whether biological validation of an algorithm may be considered to constitute a true validation of its performance. One of the reasons is incompleteness of biological knowledge. Hence, any bicluster that has been correctly depicted by a biclustering algorithm may be still erroneous, as GO/KEGG annotations or connected genes in a TRN are increasingly expanded [48].

Synthetic Datasets. Synthetic datasets may be used as benchmark as well. Most recognizable datasets have been generated by Prelic eat al. [45], Li et al. [35] and Hochreiter et al. [25]. First two benchmark datasets are small (50-100 genes), contain biclusters of equal sizes and have only simultaneous row and column overlaps. The third one, designed to match gene expression data characteristic in terms of heavy tails, contains 100 datasets of size 1000x100 with 10 multiplicative biclusters in each and additive noise. This artificial scenario is claimed to be more realistic in terms of densities and moments of datasets compared to real data [25].

4.3 Relative Indices

The third type of indices measure which recovered structure is better according to a quality measure and how input parameters influence biclustering outcome. Comparison between two different sets of biclusters may be achieved with different consensus measures [22][45][35]. Various modifications of Jaccard index are commonly used consensus measures for computing distance between two sets. Jaccard index represents the fraction of row-column combinations in both biclusters from all row-column combinations in at least one of biclusters (18):

$$f_j(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{18}$$

Those techniques in general do not take into account overlapping biclusters or entirely consider different numbers of biclusters. Novel consensus score for biclustering has been defined by [25] and bases on computing pairwise comparison between two sets of biclusters with Munkres algorithm [39], quite similar to a technique applied to visualize clustering results for proteins [44].

Another index which is worth mentioning is the F_1 index proposed by [21]. Some consider relative indices as determination of the best parametrization set for the algorithm [45][48]. The importance of proper choice of input parameters and their impact on biclustering outcome for many biclustering algorithms have been repetitively emphasized [16][17].

4.4 Conclusions

In this article current trends in biclustering are summarized with respect to proximity measures of biclusters. With many approaches available, most promising area of development of biclustering algorithms seems to be correlation based measures, especially non-parametric ones.

Novel approaches may also agglomerate results obtained from different biclustering approaches. There is a tremendous perspective for developing ensemble approach that will combine different measures and automatically detect which scheme should be applied basing on data analysis.

In the second part of the article, we classify different state-of-the-art validation techniques of biclustering algorithms. Novel approaches need to perform well in synthetic datasets in order to become recognizable, followed by successes with real data examples. So-called *reference set* of biclustering algorithms needs to be updated to include at least the following algorithms: BBC [22], CPB[8], QUBIC[35], ISA[28], FABIA[25]. Each of those algorithms achieves best results its specific cateogory of biclustering [16]. Comparison against previous algorithms such as CC [12], OPSM [6] or xMotifs[40], SAMBA [51] or HCL[15] could be treated as optional [35]. To our best knowledge, this is the first in-depth analysis considering the proximity measures fundamental in design of biclustering algorithm architecture.

Acknowledgments. The author would like to express his gratitude to prof. Krzysztof Boryczko for his consultancy and support.

References

1. Aguilar-Ruiz, J.: Shifting and scaling patterns from gene expression data. Bioinformatics 21(20), 3840–3845 (2005)
2. Alizadeh, A., Eisen, M., Davis, R., Ma, C., Lossos, I., Rosenwald, A., Boldrick, J., Sabet, H., Tran, T., Yu, X., et al.: Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. Nature 403(6769), 503–511 (2000)
3. Armstrong, S., Staunton, J., Silverman, L., Pieters, R., den Boer, M., Minden, M., Sallan, S., Lander, E., Golub, T., Korsmeyer, S., et al.: Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia. Nature Genetics 30(1), 41–47 (2002)

4. Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., et al.: Gene ontology: tool for the unification of biology. Nature Genetics 25(1), 25 (2000)
5. Ayadi, W., Elloumi, M., Hao, J.: Pattern-driven neighborhood search for biclustering of microarray data. BMC bioinformatics 13(suppl. 7), S11 (2012)
6. Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering local structure in gene expression data: the order-preserving submatrix problem. In: Proceedings of the Sixth Annual International Conference on Computational Biology, RECOMB 2002, pp. 49–57. ACM, New York (2002), http://doi.acm.org/10.1145/565196.565203
7. Bozdağ, D., Kumar, A.S., Catalyurek, U.V.: Comparative analysis of biclustering algorithms. In: Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology, BCB 2010, pp. 265–274. ACM, New York (2010), http://doi.acm.org/10.1145/1854776.1854814
8. Bozdağ, D., Parvin, J.D., Catalyurek, U.V.: A biclustering method to discover co-regulated genes using diverse gene expression datasets. In: Rajasekaran, S. (ed.) BICoB 2009. LNCS, vol. 5462, pp. 151–163. Springer, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-00727-9_16
9. Bryan, K.: Biclustering of expression data using simulated annealing. In: Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems, CBMS 2005, pp. 383–388. IEEE Computer Society Press, Washington, DC (2005), http://dx.doi.org/10.1109/CBMS.2005.37
10. Chen, G., Jaradat, S., Banerjee, N., Tanaka, T., Ko, M., Zhang, M.: Evaluation and comparison of clustering algorithms in analyzing es cell gene expression data. Statistica Sinica 12(1), 241–262 (2002)
11. Chen, P., Popovich, P.: Correlation: Parametric and nonparametric measures, pp. 137–139. Sage Publications, Incorporated (2002)
12. Cheng, Y., Church, G.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, vol. 8, pp. 93–103 (2000)
13. Choi, S., Cha, S., Tappert, C.: A survey of binary similarity and distance measures. Journal of Systemics, Cybernetics and Informatics 8(1), 43–48 (2010)
14. Dharan, S., Nair, A.S.: Biclustering of gene expression data using reactive greedy randomized adaptive search procedure. BMC Bioinformatics 10(suppl. 1), S27 (2009)
15. Eisen, M., Spellman, P., Brown, P., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences 95(25), 14863–14868 (1998)
16. Eren, K., Deveci, M., Küçüktunç, O., Çatalyürek, Ü.: A comparative analysis of biclustering algorithms for gene expression data. Briefings in Bioinformatics (2012)
17. Erten, C., Sözdinler, M.: Biclustering expression data based on expanding localized substructures. In: Rajasekaran, S. (ed.) BICoB 2009. LNCS, vol. 5462, pp. 224–235. Springer, Heidelberg (2009)
18. Faith, J., Driscoll, M., Fusaro, V., Cosgrove, E., Hayete, B., Juhn, F., Schneider, S., Gardner, T.: Many microbe microarrays database: uniformly normalized affymetrix compendia with structured experimental metadata. Nucleic Acids Research 36(suppl. 1), D866–D870 (2008)
19. Gasch, A., Spellman, P., Kao, C., Carmel-Harel, O., Eisen, M., Storz, G., Botstein, D., Brown, P.: Genomic expression programs in the response of yeast cells to environmental changes. Science Signalling 11(12), 4241 (2000)
20. Gat-Viks, I., Sharan, R., Shamir, R.: Scoring clustering solutions by their biological relevance. Bioinformatics 19(18), 2381–2389 (2003)

21. Getz, G., Levine, E., Domany, E.: Coupled two-way clustering analysis of gene microarray data. Proceedings of the National Academy of Sciences 97(22), 12079–12084 (2000)
22. Gu, J., Liu, J.S.: Bayesian biclustering of gene expression data. BMC genomics 9(suppl. 1), 4 (2008)
23. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Journal of Intelligent Information Systems 17(2), 107–145 (2001)
24. Hartigan, J.: Direct clustering of a data matrix. Journal of the American Statistical Association 67(337), 123–129 (1972)
25. Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., Kasim, A., Khamiakova, T., Van Sanden, S., Lin, D., Talloen, W., et al.: Fabia: factor analysis for bicluster acquisition. Bioinformatics 26(12), 1520–1527 (2010)
26. Hoshida, Y., Brunet, J., Tamayo, P., Golub, T., Mesirov, J.: Subclass mapping: identifying common subtypes in independent disease data sets. PloS One 2(11), e1195 (2007)
27. Ihmels, J., Bergmann, S., Barkai, N.: Defining transcription modules using large-scale gene expression data. Bioinformatics 20(13), 1993–2003 (2004)
28. Ihmels, J., Friedlander, G., Bergmann, S., Sarig, O., Ziv, Y., Barkai, N., et al.: Revealing modular organization in the yeast transcriptional network. Nature Genetics 31(4), 370–378 (2002)
29. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. 31(3), 264–323 (1999), http://doi.acm.org/10.1145/331499.331504
30. Jain, A.K., Dubes, R.: Algorithms for clustering data. Prentice-Hall, Inc. (1988)
31. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recogn. Lett. 31(8), 651–666 (2010), http://dx.doi.org/10.1016/j.patrec.2009.09.011
32. Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T., et al: Kegg for linking genomes to life and the environment. Nucleic acids research 36(suppl. 1), D480–D484 (2008)
33. Kerr, G., Ruskin, H., Crane, M., Doolan, P.: Techniques for clustering gene expression data. Computers in Biology and Medicine 38(3), 283–293 (2008)
34. Lazzeroni, L., Owen, A., et al.: Plaid models for gene expression data. Statistica Sinica 12(1), 61–86 (2002)
35. Li, G., Ma, Q., Tang, H., Paterson, A., Xu, Y.: Qubic: a qualitative biclustering algorithm for analyses of gene expression data. Nucleic Acids Research 37(15), e101–e101 (2009)
36. Liu, F., Zhou, H., Liu, J., He, G.: Biclustering of gene expression data using eda-ga hybrid. In: IEEE Congress on Evolutionary Computation, CEC 2006, pp. 1598–1602. IEEE (2006)
37. Liu, J., Li, Z., Hu, X., Chen, Y.: Biclustering of microarray data with mospo based on crowding distance. BMC bioinformatics 10(suppl. 4), S9 (2009)
38. Madeira, S., Oliveira, A.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics 1(1), 24–45 (2004)
39. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial & Applied Mathematics 5(1), 32–38 (1957)
40. Murali, T., Kasif, S.: Extracting conserved gene expression motifs from gene expression data. In: Proc. Pacific Symp. Biocomputing, vol. 3, pp. 77–88 (2003)
41. Myers, J., Well, A.: Research design and statistical analysis. Lawrence Erlbaum (2002)
42. Nepomuceno, J., Troncoso, A., Aguilar-Ruiz, J., et al.: Biclustering of gene expression data by correlation-based scatter search 4(3) (2011)

43. Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., Kanehisa, M.: Kegg: Kyoto encyclopedia of genes and genomes. Nucleic Acids Research 27(1), 29–34 (1999)
44. Orzechowski, P., Boryczko, K.: Parallel approach for visual clustering of protein databases. Computing and Informatics 29(6+), 1221–1231 (2010), http://www.cai.sk/ojs/index.php/cai/article/view/140
45. Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22(9), 1122–1129 (2006)
46. Romesburg, C.: Cluster analysis for researchers. Lulu. com (2004)
47. Roy, S., Bhattacharyya, D., Kalita, J.: Deterministic approach for biclustering of co-regulated genes from gene expression data. Advances in Knowledge-Based and Intelligent Information and Engineering Systems 243, 490–499 (2012)
48. Santamaría, R., Quintales, L., Therón, R.: Methods to bicluster validation and comparison in microarray data. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 780–789. Springer, Heidelberg (2007)
49. Sharan, R., Elkon, R., Shamir, R.: et al.: Cluster analysis and its applications to gene expression data. In: Ernst Schering Res Found Workshop, vol. 38, pp. 83–108 (2002)
50. Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Molecular Biology of the Cell 9(12), 3273–3297 (1998)
51. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics 18(suppl. 1), S136–S144 (2002)
52. Tavazoie, S., Hughes, J., Campbell, M., Cho, R., Church, G., et al.: Systematic determination of genetic network architecture. Nature Genetics 22, 281–285 (1999)
53. Teng, L., Chan, L.: Discovering biclusters by iteratively sorting with weighted correlation coefficient in gene expression data. Journal of Signal Processing Systems 50(3), 267–280 (2008)
54. Wilcox, R.: Introduction to robust estimation and hypothesis testing. Academic Press (2005)
55. Wille, A., Zimmermann, P., Vranová, E., Fürholz, A., Laule, O., Bleuler, S., Hennig, L., Prelic, A., Von Rohr, P., Thiele, L., et al: Sparse graphical gaussian modeling of the isoprenoid gene network in arabidopsis thaliana. Genome Biol. 5(11), R92 (2004)
56. Yang, J., Wang, H., Wang, W., Yu, P.: Enhanced biclustering on expression data. In: Proceedings of Third IEEE Symposium on Bioinformatics and Bioengineering, pp. 321–327 (March 2003)
57. Yip, K.Y., Cheung, D.W., Ng, M.K.: Harp: A practical projected clustering algorithm. IEEE Trans. on Knowl. and Data Eng. 16(11), 1387–1397 (2004), http://dx.doi.org/10.1109/TKDE.2004.74

Clustering and Classification of Time Series Representing Sign Language Words

Mariusz Oszust and Marian Wysocki

Department of Computer and Control Engineering,
Rzeszow University of Technology,
2 Wincentego Pola, 35-959 Rzeszow, Poland
{moszust,mwysocki}@kia.prz.edu.pl
http://www.kia.prz.edu.pl

Abstract. The paper considers time series with known class labels representing 101 words of Polish sign language (PSL) performed many times in front of a camera. Three clustering algorithms: K–means, K–medoids and Minimum Entropy Clustering (MEC) are compared. Preliminary partitioning of the data set is performed with help of immune based optimisation. Some time series representations and different clustering quality indices are considered. It is shown that clustering is able to reveal existing natural division. Moreover, it gives an opportunity to learn the issues of processing large number of multidimensional data and to identify potential problems which may occur in automatic classification of signed expressions. Results of ten–fold cross–validation tests for nearest neighbour classification are also given.

Keywords: clustering of time series, dynamic time warping, cluster validation, immune based optimisation, sign language recognition, computer vision.

1 Introduction

Nowadays, there is an increasing interest in natural human–computer interfaces [20], and among other things interfaces for hearing impaired people. In order to build a system which offers gesture interpretation at first we need to solve a problem of data acquisition. Many systems use accelerometers [16] or data gloves [30] as primary source of information about hands' positions, orientations and shapes. Systems of this kind are very accurate but may hinder the freedom of hands and affect the performed gestures. In addition, gloves have a fixed size and are not always available. A different approach consists in using electromyography (EMG) [16] but here, in turn, we need to take into account the effect of fatigue in the classifier construction [10].

Therefore, many researches use computer vision for hand detection and tracking [23]. In order to facilitate the segmentation of the hand from the background some authors use coloured hand gloves or markers [4]. Many works are based on detection of skin-coloured objects [23]. After segmentation hands are tracked and then selected features are applied to train a classifier.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 218–229, 2013.

Two main approaches to gesture classification apply different variations of neural networks (NN) [35] or hidden Markov models (HMM) [26], but one can also find works with boosting [6] or dynamic time warping (DTW) [3,31].

Since construction of a classifier is often complex and time–consuming, it is worth examining a data set in advance to discover some future problems. Clustering and validation of the results can reveal existing natural divisions in the data set representing gestures, can show issues related to the processing of large number of multidimensional data and identify problems that may occur in recognition. The approach presented in the paper refers to time series clustering [34], choosing clustering validity index [21,34] and sign language recognition with DTW [24,31]. The contribution of the paper lies in (1) evaluation of time series clustering methods applied to real data with known class labels representing sign language words, (2) taking into consideration various data representations, (3) discussion of cluster validation indices, (4) conducting classification experiments for various data representations. According to our best knowledge, there is no previous research concerning the clustering of feature vectors representing sign language gestures in order to evaluate some potential problems, which may occur in recognition. Usually, the researchers do not examine processed gesture samples for mutual similarities, leaving the reader without the possibility to judge the difficulty of recognition tasks.

The rest of the paper is organized as follows. Section 2 gives preliminary information concerning PSL, the features used in the following chapters, clustering algorithms and clustering quality indices. Section 3 covers experiments involving clustering of isolated PSL words, validation of clustering results and recognition. Section 4 concludes the paper.

2 Preliminary Information

2.1 Characteristics of the PSL

Number of people with hearing loss in Poland can be up to several hundred thousand. For the most of them - about 50000 [25] with severe and profound hearing loss - sign language is the main mean of communication. Sign language is the language of deaf people communicated by their deaf parents. PSL is a distinguishing feature of the Polish community of deaf culture [27].

PSL signs are static or dynamic. Most of them are two–handed. For one–handed signs the so–called dominant hand performs the sign, whereas for two–handed signs the non–dominant hand is also used. The hands often touch each other or appear against the background of the face.

Every sign can be analysed by specifying at least three components [14]: (i) the place of the body against which the sign is made, (ii) the shape of a hand or hands, (iii) the movement of a hand or hands. Although in practical sign language communication some additional features (such as lip shape, face expression, etc.) are often used, we do not consider them in this paper.

2.2 Construction of the Feature Vectors

We use seven features for the right hand and the same features for the left hand: the position of the hand with respect to the face (three spatial coordinates), the area, orientation, compactness and eccentricity of a hand [33]. Feature vectors were extracted by image processing techniques. For detection of the signer's hands and face we used a method based on a chrominance model of human skin in the form of a 2D Gaussian distribution in the normalised RGB space. To detect skin–toned regions in a colour image, the image was transformed into a gray–tone form using the skin colour model and thresholded. The areas of the objects toned in skin colour, their centres of gravity and ranges of motion were analysed to recognise the right hand, the left hand and the face. Comparison of neighbouring frames helped to notice whether the hands (the hand and the face) touched or partially covered each other. In order to ensure correct segmentation there were some restrictions for the background and the clothing of the signer.

2.3 Clustering Techniques

The aim of clustering is to group vectors into sets on a basis of their resemblance [26,34]. In most cases there is no information about how vectors should be decomposed i.e. it is difficult to estimate the true number of classes and the best approach to cluster them.

Time series is a vector containing chronologically ordered observations [12,15], and it has to be considered as a whole. If we want to place vectors into subsets with similar elements we have to take into account continuous character of time series and chose appropriate technique for their comparison. Due to this reason time series clustering differs from high–dimensional data clustering, even if time series are equal in length [19]. Most clustering algorithms calculate similarity between vectors assuming equal dimensions of data being clustered (series length is considered as a dimension). The most common approach to reduce time series length is resampling, finding characteristic points (perceptually important points, PIP), compressing by principal component analysis (PCA) or singular value decomposition (SVD) [12]. Unfortunately, these solutions can produce time series which have different lengths.

Another solution offering representation of time series considers each time series as a vector of pairwise proximities (distances) [34] between it and every other time series in the data set. Such long vectors of equal lengths are grouped in a proximity matrix. Proximity matrices are also successfully applied in classification tasks [5]. To compute similarity between time series one can apply discrete wavelet transformation (DWT) [29] or dynamic time warping (DTW) [34].

After determining proximity matrix of DTW distances between time series representing sign language gestures one obtains vectors of length equal to the number of gestures. Comparison of such long vectors during clustering can create problems (curse of dimensionality) [26,34]. One can apply common ideas for dimensionality reduction like aforementioned PCA. PCA is a procedure of

transforming a set of correlated variables into a set of uncorrelated variables called principal components [34]. This transformation is defined in such a way that the first principal components have the main account of the variability in the data. After rejecting less important components one obtains low dimensional vectors. It is worth mentioning that in this approach we process available gesture data in a single batch. When a new PSL word is joined we would require to update PCA projections in a sequential matter using an online approach to PCA estimation [32].

In our experiments we used three clustering algorithms [34] K–means, K–medoids and MEC [18]. In general, each technique starts from an initial assignment of elements into a given number of clusters and iteratively improves the assignments, trying to minimise its inner clustering criterion. If no further changes in assignments appear, the algorithm stops. The strategy does not in general produce a globally optimal solution, but nonetheless it may yield locally optimal solutions.

In the K-means technique the criterion is the mean distance among the elements and their cluster centres, i.e.

$$J(D) = \frac{1}{n} \sum_{i=1}^{m} \sum_{z \in C_i(D)} d(z, \bar{C}_i(D)) \tag{1}$$

where D denotes a decomposition of the data set, z is an element of the cluster $C_i(D)$, $\bar{C}_i(D)$ is the center of this cluster, $d(.,.)$ denotes a distance between the elements, n is number of elements and m is number of clusters. The K–medoids has similar criterion with the difference that it uses the distance between elements in a cluster and its medoid. The MEC algorithm, in turn, uses entropy measured on a posteriori probabilities as the criterion of clustering. In fact, it is the conditional entropy of clusters given the observations. Therefore two problems are addressed in this technique: (1) estimating a posteriori probabilities and (2) minimizing the entropy. Experiments presented in [18] show that MEC performs significantly better than K–means clustering, hierarchical clustering, SOM and EM. Moreover, it is able to correctly reveal the structure of data and identify outliers simultaneously. In our work we used the Java package prepared by the authors of the algorithm [22]. It is worth mentioning that MEC starts from the solution obtained by one iteration of K–means algorithm and improves it. The choice of number of clusters is usually made experimentally or applying clustering quality indices. In our problem the number of clusters was known and equal to the number of gestures' classes.

2.4 Clustering Quality Indices

For clustering evaluation we applied the following indices [21]: Dunn index, Davies–Bouldin index, I index and RAND index. These indices are often used for validation of correct number of clusters. We are interested in their ability to indicate quality of clustering.

The Dunn index DI [34], [21] is defined by two parameters: the diameter $diam(C_i)$ of the cluster C_i and the set distance $\delta(C_i, C_j)$ between C_i and C_j, where:

$$diamC_i = max_{x,y \epsilon C_i}\{d(x,y)\}, \delta(C_i, C_j) = min_{x \epsilon C_i, y \epsilon C_j}\{d(x,y)\} \tag{2}$$

and $d(x, y)$ indicates the distance between points x, y.

$$DI = min_{1 \leq j \leq m}\{min_{1 \leq i \leq m, i \neq j}\{\frac{\delta(C_i, C_j)}{max_{1 \leq k \leq m} diamC_k}\}\} \tag{3}$$

Larger values of DI correspond to good clusters.

Davies–Bouldin index DB attempts to maximise the between cluster distance $S(C_i, C_j)$ while minimising the average distance $S_m(C_i)$ between the cluster centroid and the other points.

$$DB = \frac{1}{m}\sum_{i=1}^{m} min_{i \neq j}\{\frac{S_m(C_i) + S_m(C_j)}{S(C_i, C_j)}\} \tag{4}$$

where m is number of clusters. Less value of DB index indicates better clusters.

The I index is defined as follows:

$$I(m) = (\frac{1}{m} \times \frac{E_1}{E_m} \times D_m)^P \tag{5}$$

where $E_m = \sum_{i=1}^{m}\sum_{j=1}^{n} u_{ij} S(C_i, x_j)$, $D_m = max_{i,j=1}^{m} S(C_i, C_j)$, $S(C_i, x_j)$ is distance between x_j-th object and C_i cluster centre, u_{ij} denotes the membership coefficient for x_j, and the power P is used to control the contrast between the different cluster configurations. The I index is useful for determining the correct number of clusters, which is chosen when $I(m)$ is maximal. In our case we set m value equal to the expected number of clusters.

The RAND index measures the number of pairwise agreements between two decompositions D_1 and D_2. This index is useful when we know how data should be clustered, say decomposition D_1.

$$RAND(D_1, D_2) = \frac{a+b}{a+b+c+d} \tag{6}$$

where a denotes the number of pairs of elements that are in the same cluster in both decompositions, b denotes the number of pairs of elements that are in different clusters in both decompositions, c denotes the number of pairs of elements that are in the same cluster in the first decomposition D_1, but in different clusters in D_2, and d denotes the number of pairs of elements that are in different clusters in D_1 but in the same cluster in D_2.

Before clustering we must define a distance between sequences. We used DTW [34], which allows a nonlinear mapping of one sequence to another by minimizing a distance between them. The main motivation for using DTW is its ability to expand or compress the time when comparing sequences that are similar but

locally out of phase. For example, some related parts of gestures representing the same expression may be performed with different velocities. Lengths of compared sequences can be different.

Initial partitioning of data into clusters has crucial influence on final clustering, therefore we run experiments with an optimisation algorithm, which proposes initial partitioning leading to better clustering results. In order to solve this problem we applied the immune algorithm CLONALG [8,28].

The main loop of the algorithm is repeated *gen* times, where *gen* is the number of generations. It consists of four main steps: one initial step where all the elements of the population are evaluated and three transformation steps: clonal selection, hipermutation, apoptosis. Elements in the population are often called lymphocytes or antibodies.

1. Evaluation. For each element D in the population P compute $J(D)$ and perform ordering of the elements.
2. Clonal selection. Choose a reference set $P_a \subset P$ consisting of h elements at the top of the ranking obtained in step 1.
3. Hipermutation.
   (a) For each $D \in P_a$ make c mutated clones $Dc_j, j = 1, 2, \ldots, c$, compute their value $J(Dc_j)$ and place the clones in the clonal pool CP.
   (b) Order the elements of $P \cup CP$, choose a subset $P_c \subset P \cup CP$ containing B best elements, where B denotes the size of P.
4. Apoptosis. Replace b worst elements in P_C by randomly generated elements.
5. Set $P = P_C$.

In the algorithm the current population P is mixed with the clonal pool CP and the predefined number of best elements (i.e. at the top of the ranking) is picked up to form a new population. The last step of the main loop replaces b worst solutions by randomly generated elements. In our problem each element of population P represents one decomposition of the whole set of time series. Hipermutation process consists of a random choice of two clusters, draw two elements and changing their assignments.

3 Experiments

In this Section we present the results of experiments based on real sequences obtained for signed Polish words. The sequences represent 101 words, which can be used at the doctor's and in the post office. Each is characterised by a vector of 14 features mentioned in Subsection 2.2. We used a data set consisting of sequences of feature vectors for 40 realisations of each of the 101 words. Gestures have been performed by two signers. One person is a PSL teacher, the other has learnt PSL for the purposes of this research. Each signer repeated each word 20 times. The data have been registered with the rate 25 frames/s.

The first experiment concerns comparison of clustering quality indices for PSL words. Mean values of quality indices computed for 100 executions of the clustering procedure (due to random assignment of elements to the clusters at the

beginning of the clustering) are shown in Table 1. Appropriate inner clustering criteria of each clustering algorithm were computed and presented as well. The table shows four different approaches: (i) evaluation of clustering based on ground truth, also with clustering which starts from such division; (ii) clustering which starts from randomly assigned elements to the clusters; (iii) randomly started clustering with similarity vectors reduced by PCA and (iv) clustering results improved by immune algorithm CLONALG. Values of indices to be maximised were multiplied by -1 for the simplicity of comparison (smaller values mean better quality now). Best values are marked in bold. For algorithms with random start we also observed the time of execution. Experiments were run on the processing unit with 3.3 GHz, 4 cores, 16GB RAM, Windows 7 64 bit, and Java.

Each clustering technique which starts from ground truth cluster assignment improves its quality criterion, which not always means improving values of cluster quality indices. Such clustering often leads to better clusters than a clustering starting from random assignment. Among clustering techniques K–medoids is characterised by short computation time but is seldom evaluated as producing good clustering. MEC in conjunction with dimensionality reduction by PCA mostly gives the best quality of clusters. The use of vectors reduced by PCA (length equal to 30 instead of 4040) significantly reduces computation time when clustering algorithms operate on vectors (K–means and MEC). The MEC algorithm is working with neighbours defined for each dimension, and K–means algorithm calculates multidimensional mean elements for each cluster, therefore the number of dimensions strongly influences on execution time for the both of them. When comparing the execution time of algorithms (Table 1) we can see that K–medoids algorithm is the fastest, but in the case of PCA based representation it also requires time consuming preparation of the proximity matrix. Obtained computation time could be shortened by changing programming language or implementation of algorithms, e.g. K–means could be improved using coresets [11], the triangle inequality [9] or by bootstrap averaging [7]. It can be noted that each clustering algorithm has changed the initial partitioning, marked as known division. Nonetheless a value of each inner clustering index obtained by starting the algorithms from known division turned out to be better than by starting from random division.

Values of clustering validity indices obtained in experiments do not favour any index or methods of clustering. Among all algorithms MEC was the best in the majority of experiments. RAND index values show that MEC in all cases grouped data in a way that could be considered as the closest to the known division of PSL words. Therefore, despite the long execution time in comparison with K–medoids, MEC seems to find the natural division of the analysed data.

We used RAND index (6) to compare clustering results with the ground truth cluster assignment. The number of agreements $(a + b)$ turned out to be much larger than the number of disagreements $(c + d)$. Therefore resulting RAND values are very similar for all clustering results.

Table 1. Mean values of clustering quality indices of 100 experiments. For convenience indices being maximised were multiplied by -1, RAND values remain without changes. As *KD* we marked results with initial data division consistent with known division of gestures' executions into their classes. Fitness function in immune algorithm is a quality measure used by a given algorithm, CLONALG's parameters were as follows: $B = h = 10$, $c = 5$, $b = 1$, $gen = 10$.

Algorithm	Time [ms]	DB	I	Dunn	K-means	MEC	K-medoids	RAND
KD	-	2,24	-3,05E-09	-0,039	7,675	0,0365	0,0786	1
KD + K-means	39874	2,11	-3,25E-09	-0,003	4,432	0,0584	0,1034	0,9902
KD + K-medoids	187	1,66	-3,99E-09	-0,01	7,594	0,0478	0,0754	0,9984
KD + MEC	2776	2,25	-3,05E-09	-0,039	7,546	0,0217	0,0789	0,9997
K-means	66772	2,39	-3,55E-09	-0,004	4,507	0,0648	0,1149	0,9863
K-medoids	233	4,55	-2,52E-09	-0,006	15,777	0,1491	0,1217	0,9839
MEC	50255	2,3	-3,60E-09	-0,005	4,53	0,0329	0,1147	0,9863
K-means + PCA	386	1,27	-3,90E-06	-0,016	4,501	0,0668	0,116	0,9862
K-medoids + PCA	183	1,57	-3,26E-06	-0,005	7,454	0,1323	0,1373	0,981
MEC + PCA	710	1,26	-3,88E-06	-0,028	4,521	0,0481	0,1154	0,9862
CLONALG + K-means	-	2,44	-4,62E-09	-0,005	4,218	0,0619	0,1136	0,9862
CLONALG + K-medoids	-	3,93	-3,52E-09	-0,007	12,7	0,1287	0,1093	0,9865
CLONALG + MEC	-	2,22	-3,78E-09	-0,006	4,528	0,0230	0,1143	0,9861
CLONALG + K-means + PCA	-	2,36	-4,49E-09	-0,005	4,211	0,0646	0,1128	0,9861
CLONALG + K-medoids + PCA	-	2,54	-2,22E-09	-0,004	8,306	0,1249	0,1291	0,9830

We can also note apparent improvement of clustering quality indices when CLONALG proposes initial data division. It is mostly evident for MEC, where obtained results are close to the known division of data. K–medoids was not able to produce well evaluated clusters; that is why we have run another experiment, in which optimisation was divided into five stages. Results of each stage were improved in the next one. It reduces time-consumption and allows to obtain better results than a single experiment with one population. After 500 generations the value of K–medoids quality criterion turned out to be better than the value obtained for both natural division of data and its improvement by K–medoids.

By analysing the data obtained after the fifth stage for K–medoids we noticed that some expressions were assigned to different clusters and some dominated in clusters. For example the word *lie down* did not dominate in any cluster

(some of them were assigned with word *on*) and expression *inflammation* dominated in two clusters. Most clusters were consistent with the natural division. As similar experiment with MEC algorithm (also 5 stages) resulted in clusters with some words executions which could lead to misclassification errors during recognition, e.g.: *flue* with *throat*, *tooth* with *teeth*, *lie down* with *on* and *hearing* with *head* (Fig. 1). We also performed research using algorithms designed to

Fig. 1. Consecutive frames of exemplary execution of word *hearing* (top) and word *head* (bottom)

process multidimensional data [17]. Unfortunately, popular subspace clustering algorithms implemented in ELKI library [1] (CLIQUE and SUBCLUE) needed too much computation time or ended computation with the depletion of memory resources. Similar problems were met in work [13]. Density–based clustering algorithms, i.e. starting from the estimated density distribution of corresponding nodes (DBSCAN and OPTICS) [2] found inappropriate number of clusters: DBSCAN about 200 and OPTICS only 80. Both algorithms are characterised by parameters which are difficult to set, especially a minimal number of points forming a cluster. Computation took from 180 s to 2400 s.

Another experiment concerns recognition of sign language expressions using nearest neighbour approach [34] and earlier proposed data representation (DTW distances, vector of DTW distances or short vectors after processing with PCA). In the experiment we used cross–validation dividing the data set into ten disjoint subsets. Each subset consisted of data corresponding to four repetitions of each word (two repetitions performed by each signer). We performed ten experiments using nine subsets as the training set and the remaining subset as the test set. The results presented in Table 2 confirm that our sequences comparison technique (DTW) and data representation as the vector of distances are suitable for sign expressions representation. It means that feature vector derived from video material has discriminative properties. Even its transformation to the vector of DTW distances and then shorter vectors produced by PCA still preserve enough information to correctly recognise sign language expressions. This experiment also justified our choice of time series representation as vectors of pairwise DTW distances between them.

Clustering validation gave slight knowledge about which gesture execution could lead to classification errors. Such a comparison shows how similar are gesture executions from different classes.

Table 2. Ten–fold cross–validation test for nearest neighbour classifier and different time series representations. The results are given in %.

Cross–validation variants									
1	2	3	4	5	6	7	8	9	10
Time series + DTW, mean recognition rate - **98,99**%, stdev - 1,31%									
95,30	99,50	99,50	99,01	99,50	99,26	99,26	99,26	99,75	99,50
DTW proximity matrix + Euclidean distance, mean recognition rate - **96,01**%, stdev - 2,91%									
88,12	95,05	96,53	98,27	97,77	96,53	96,78	97,28	96,53	97,28
DTW proximity matrix + PCA + Euclidean distance, mean recognition rate - **95,40**%, stdev - 3,34%									
86,39	94,55	95,30	98,02	97,52	95,79	96,53	97,28	95,79	96,78

Wrongly recognised executions from several variants of cross–validation (first row in Table 2) were earlier pointed out in clustering validity tests, e.g. problems with words such as: *lay down*, *hearing* or *teeth*.

4 Conclusions

Clustering of time series with known class labels representing 101 words of Polish sign language has been considered. Three clustering algorithms: K–means, K–medoids and Minimum Entropy Clustering (MEC) have been used and compared in terms of quality of resulting clusters and their compatibility with natural data division. In general, the algorithms find local solutions depending on initial partitioning of the data set. Therefore, the immune based optimisation algorithm CLONALG has been applied. Clustering validation proved to be helpful in identification of problems that could occur during recognition.

Some time series representations and different clustering quality indices were considered. It is shown that clustering is able to reveal the existing natural division. Moreover, it gives an opportunity to learn the issues of processing large number of multidimensional data and to identify potential problems which may occur in automatic classification of signed expressions.

Results obtained for the MEC algorithm turned out to be closest to the natural data division, but K–medoids algorithm yielded the shortest execution time.

The ten–fold cross–validation tests for the nearest neighbour classification confirm that using time series of feature vectors of sign language words is suitable sign data representation. Time series transformation to the vector of DTW distances and even shorter vectors produced by PCA still preserve enough information to correctly recognise sign language expressions.

References

1. Achtert, E., Bernecker, T., Kriegel, H.-P., Schubert, E., Zimek, A.: Elki in time: Elki 0.2 for the performance evaluation of distance measures for time series. In: Mamoulis, N., Seidl, T., Pedersen, T.B., Torp, K., Assent, I. (eds.) SSTD 2009. LNCS, vol. 5644, pp. 436–440. Springer, Heidelberg (2009)
2. Ankerst, M., Breunig, M.M., Kriegel, H.-P., Sander, J.: Optics: ordering points to identify the clustering structure. SIGMOD Record 28(2), 49–60 (1999)
3. Athitsos, V., Neidle, C., Sclaroff, S., Nash, J., Stefan, R., Thangali, A., Wang, H., Yuan, Q.: Large lexicon project: American sign language video corpus and sign language indexing/retrieval algorithms (2010)
4. Awad, G., Han, J., Sutherland, A.: Novel boosting framework for subunit–based sign language recognition. In: Proceedings of the 16th IEEE International Conference on Image Processing, ICIP 2009, pp. 2693–2696. IEEE Press, Piscataway (2009)
5. Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L.: Similarity–based Classification: Concepts and Algorithms. Journal of Machine Learning Research, 747–776 (2009)
6. Cooper, H.: Sign Language Recognition: Generalising to More Complex Corpora. Centre For Vision Speech and Signal Processing. PhD thesis, University Of Surrey (2010)
7. Davidson, I., Satyanarayana, A.: Speeding up k-means Clustering by Bootstrap Averaging. In: Proceedings of the IEEE ICDM Workshop on Clustering Large Data Sets, pp. 16–25 (2003)
8. de Castro, L.N., Von Zuben, F.J.: Learning and optimization using the clonal selection principle. IEEE Transactions on Evolutionary Computation 6(3), 239–251 (2002)
9. Elkan, C.: Using the Triangle Inequality to Accelerate k-Means. In: ICML, pp. 147–153. AAAI Press (2003)
10. Flasinski, M., Myslinski, S.: On the use of graph parsing for recognition of isolated hand postures of Polish sign language. Pattern Recognition 43(6), 2249–2264 (2010)
11. Frahling, G., Sohler, C.: A fast k–means implementation using coresets. In: Symposium on Computational Geometry, pp. 135–143 (2006)
12. Fu, T.-C.: A review on time series data mining. Engineering Applications of Artificial Intelligence 24(1), 164–181 (2011)
13. Hein, A., Kirste, T.: Unsupervised detection of motion primitives in very high dimensional sensor data. In: Gottfried, B., Aghajan, H.K. (eds.) Proceedings of the 5th Workshop on Behaviour Monitoring and Interpretation, BMI 2010, Karlsruhe, Germany. CEUR Workshop Proceedings (2010)
14. Hendzel, J.K.: Polish Sign Language Dictionary. Publishing House Pojezierze (1986) (in polish)
15. Keogh, E., Lin, J.: Hot sax: Efficiently finding the most unusual time series subsequence. In: Fifth IEEE International Conference on Data Mining, pp. 226–233 (2005)
16. Kosmidou, V., Petrantonakis, P., Hadjileontiadis, L.J.: Enhanced sign language recognition using weighted intrinsic-mode entropy and signer's level of deafness. IEEE Transactions on Systems, Man, and Cybernetics, Part B 41(6), 1531–1543 (2011)
17. Kriegel, H.-P., Kröger, P., Zimek, A.: Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM The Transactions on Knowledge Discovery from Data 3(1), 1–58 (2009)

18. Li, H., Zhang, K., Jiang, T.: Minimum entropy clustering and applications to gene expression analysis. In: Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference, CSB 2004, Washington, DC, pp. 142–151 (2004)
19. Liao, W.T.: Clustering of time series data a survey. Pattern Recognition 38(11), 1857–1874 (2005)
20. Lin, D.J., Le, V., Huang, T.S.: Human–computer interaction. Visual Analysis of Humans, pp. 493–510 (2011)
21. Maulik, U., Bandyopadhyay, S.: Performance evaluation of some clustering algorithms and validity indices. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(12), 1650–1654 (2002)
22. MEC: Minimum entropy clustering Java package, `http://www.cs.ucr.edu/~hli/mec/` (accessed August 10, 2012)
23. Ong, S.C.W., Ranganath, S.: Automatic sign language analysis: A survey and the future beyond lexical meaning. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(6), 873–891 (2005)
24. Oszust, M., Wysocki, M.: Modelling and recognition of signed expressions using subunits obtained by data–driven approach. In: Ramsay, A., Agre, G. (eds.) AIMSA 2012. LNCS, vol. 7557, pp. 315–324. Springer, Heidelberg (2012)
25. Szczepankowski, B.: Sign language at School. WSiP (1988) (in polish)
26. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn. Academic Press (2008)
27. Tomaszewski, P.: Visual phonology of Polish Sign Language. Publishing House Matrix (2010) (in polish)
28. Trojanowski, K., Wierzchon, S.: Immune-based algorithms for dynamic optimization. Information Sciences 179(10), 1495–1515 (2009)
29. Tseng, V.S., Chen, C.-H., Huang, P.-C., Hong, T.-P.: Cluster-based genetic segmentation of time series with DWT. Pattern Recognition Letters 30(13), 1190–1197 (2009)
30. Vogler, C., Metaxas, D.N.: Toward scalability in ASL recognition: Breaking down signs into phonemes. In: Braffort, A., Gibet, S., Teil, D., Gherbi, R., Richardson, J. (eds.) GW 1999. LNCS (LNAI), vol. 1739, pp. 211–224. Springer, Heidelberg (2000)
31. Wang, Q., Chen, X., Zhang, L.-G., Wang, C., Gao, W.: Viewpoint invariant sign language recognition. Computer Vision and Image Understanding 108(1-2), 87–97 (2007)
32. Warmuth, M.K., Kuzmin, D.: Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. Journal of Machine Learning Research 9, 2287–2320 (2008)
33. Wysocki, M., Kapuscinski, T., Marnik, J., Oszust, M.: Vision–based Hand Gesture Recognition. Rzeszow University of Technology Publishing House (2011) (in polish)
34. Xu, R., Wunsch, D.: Clustering. Wiley-IEEE Press (2009)
35. Zahedi, M., Manashty, A.R.: Robust sign language recognition system using ToF depth cameras. CoRR (2011)

Evaluation of the Mini-Models Robustness to Data Uncertainty with the Application of the Information-Gap Theory

Marcin Pluciński

Faculty of Computer Science and Information Systems,
West Pomeranian University of Technology,
Żołnierska 49, 71-062 Szczecin, Poland
mplucinski@wi.zut.edu.pl

Abstract. The paper describes a new method based on the information-gap theory which enables an evaluation of the mini-models robustness to a specified kind of uncertainty in the data. There are presented concepts of a robustness and an opportunity of mini-models and calculations of these concepts were performed for a simple 1-D data set and next, for a more complicated 6-D data set. In both cases the method worked correctly and enabled evaluation of the robustness and the opportunity for a given lowest acceptable quality r_c or a windfall quality r_w. Additionally the method enabled choosing of the most robust model for a given level of an uncertainty.

Keywords: Mini-model, local regression, data uncertainty, information-gap theory, interval analysis.

1 Introduction

Measurement errors are a natural property of data, so if uncertain data are used for a model creation it can be important to evaluate the robustness to uncertainty of such model.

Memory-based learning methods are very often attractive approach in comparison with creating of global models based on a parametric representation. In some situations (for example: small number of samples), building of global models can be difficult and memory-based methods become then one of possible solutions for the approximation task.

Memory-based methods are very well explored and described in many bibliography positions. One of the simplest, for example, is the k-nearest neighbors method (kNN) [5, 7–9]. Another approach can be methods based on a locally weighted learning [1, 5]. Methods widely applied in this category are probabilistic neural networks and generalised regression networks [15, 18].

The concept of mini-models was introduced by prof. Andrzej Piegat. In papers [12, 13] there were described local regression models based on simplexes. Described models were linear and a research work was made only for problems in

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 230–241, 2013.

a 1 and 2-dimensional input space. In this paper, there are applied mini-models based on hyper-spheres described in [16]. Such mini-models can be linear or nonlinear and have no limitations for the problem input space dimension.

2 Linear and Nonlinear Mini-Models

The mini-model is a local regression and the answer for the question point $\mathbf{x}^*$ is calculated on a base of a local model created for k-nearest neighbors. The mini-model is always created in time of answer calculations (it means that as in other memory methods samples are only memorised during learning and the real mini-model is created only when the output is calculated for the given question point $\mathbf{x}^*$) [16].

In the simplest case the linear mini-model can be applied, and then the answer is calculated on the base of the linear regression:

$$f(\mathbf{x}^*) = \mathbf{w}^T \cdot \mathbf{x}^* \,, \tag{1}$$

where: $\mathbf{w}$ – the vector of linear mini-model coefficients found for k-neighbors.

In papers [12, 13] there are described mini-models created for sectors of the input space that have a triangle shape (in a 2-dimensional input space) or a simplex shape in a multi-dimensional input space. Such sector will be called a mini-model base. In this paper, mini-models will be created for a circular base in a 2-dimensional input space or a spherical (hiper-spherical) base in a 3 or multi-dimensional input space.

The mini-model base has a center in the question point $\mathbf{x}^*$ and its radius is defined by a distance between the point $\mathbf{x}^*$ and the most distant point from k neighbors, Fig. 1.

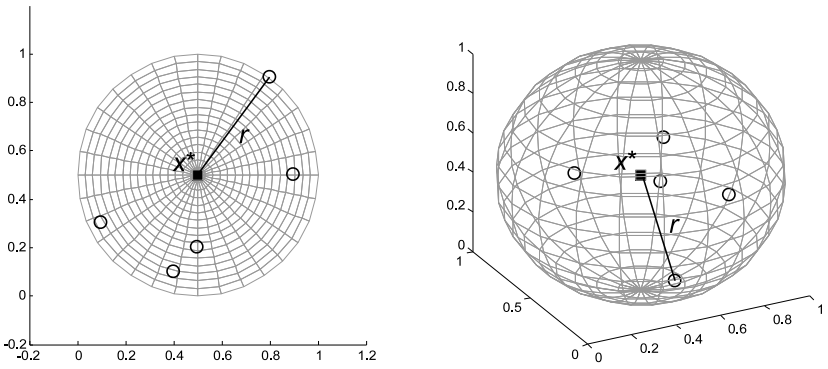


Fig. 1. The mini-model base for a 2 and 3-dimensional input space

Nonlinear mini-models have better possibilities of fitting to samples. An answer of such model is a sum of a linear mini-model and an additional nonlinear component:

$$f(\mathbf{x}^*) = \mathbf{w}^T \cdot \mathbf{x}^* + f_N(\mathbf{x}^*)\,. \tag{2}$$

As the mini-model is usually created for a small number k of nearest neighbors, the nonlinear function f_N should have a possibility of changing its shape thanks to as small number of coefficients as possible (because $N+1$ coefficients must be tuned in the vector $\mathbf{w}$ in a N-dimensional input space).

Among many inspected functions, very advantageous properties has the function:

$$f_N(\mathbf{x}) = w_N \cdot \sin\left[\frac{\pi}{2} - ||\mathbf{x} - \mathbf{x}^*||\frac{\pi}{r}\right]\,, \tag{3}$$

where: r is the radius of the mini-model base. In such created function we have only one coefficient w_N to learn. During learning we must find such a w_N value to obtain the best fit of the mini-model to k neighbors. Exemplary linear and nonlinear mini-models in a 1 and 2-dimensional input space are presented in Fig. 2 and Fig. 3.

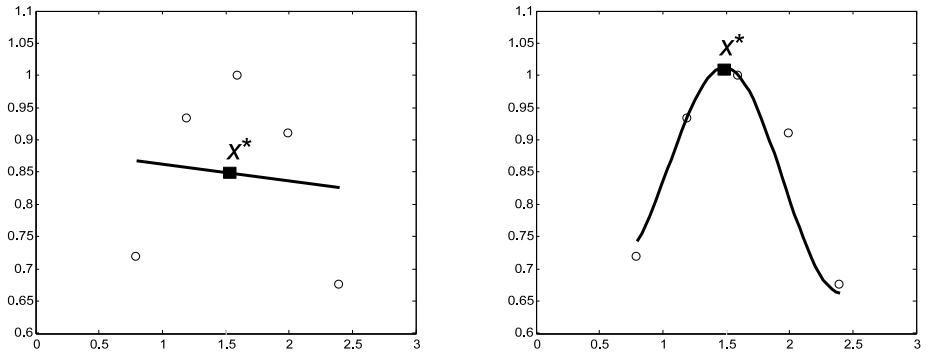


Fig. 2. Exemplary linear and nonlinear mini-model in a 1-dimensional input space

One of the main mini-model parameters is the number of neighbours k that are taken into account. It can be constant for entire data set, but in some approaches it can be dynamically varied – according to the question point location in the input space. The popular technique of k evaluation is applying 'leave one out' crossvalidation or applying two distinct data sets: training data – which is memorised, and testing data – to evaluate the real model error. The best k value is the value that gives the lowest test or crossvalidation error. The lowest test or crossvalidation error guarantees the lowest real error of the model and the best generalisation.

The approximation function based on mini-models have many useful properties [16]. It has a good accuracy and a very advantageous extrapolation features. Learning of the approximator is fast and not computationally complex.

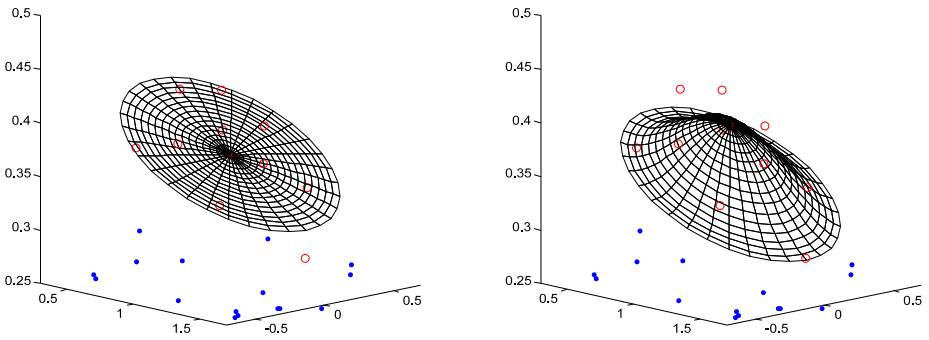


Fig. 3. Exemplary linear and nonlinear mini-model in a 2-dimensional input space

3 Model of Uncertainty

There are a lot of techniques allowing taking data uncertainty (both input and output one) into consideration. Among them we can mention methods that analyse data from a probabilistic point of view [5, 7, 19]. Other methods apply the theory of fuzzy sets [6, 11]. In this paper there will be applied the information-gap model of uncertainty based on the theory described in [2–4, 14] and the interval analysis [10].

The information-gap theory enables prediction of bounded worst case errors in the presence of a specified level of uncertainty in the input and output data. The method also enables a discrimination between various models if there's need to find the model that is the most robust to data uncertainty.

3.1 Interval Analysis

In the paper, there will be analysed data that were made uncertain by applying an interval expansion of size α in all dimensions of the data set (both in input and output attributes) [10]. The α parameter describes the unknown horizon of uncertainty in the information-gap model (described in the next subsection). After expansion each attribute becomes interval number [10] and can be defined as an ordered pair of real numbers $[a, b]$ with $a < b$ such that:

$$[a, b] = \{x : \ a \leq x \leq b\} . \tag{4}$$

During expansion each crisp attribute x is replaced by the interval $[\underline{x}, \overline{x}]$ where $\underline{x}$ represents the lower interval bound and $\overline{x}$ the upper interval bound:

$$[\underline{x}, \overline{x}] = [x - \alpha, x + \alpha] . \tag{5}$$

The special interval number arithmetic is described in details in [10]. With its application it is possible to apply mini-models after some small modifications.

First of all, the question point $\mathbf{x}^*$ and memorised data are interval values now, so the method of a calculation of a distance between them must be determined to find k neighbours. The distance between intervals can be calculated as [10]:

$$\mathrm{dist}(v_1, v_2) = \max\{|\underline{v_1} - \underline{v_2}|, |\overline{v_1} - \overline{v_2}|\}\,.$$

Finally, the distance between samples can be calculated as follow:

$$\mathrm{dist}(\mathbf{x}, \mathbf{x}^*) = \left(\sum_{j=1}^{N} \mathrm{dist}^2(x_j, x_j^*)\right)^{1/2},$$

where x_j is the j-th attribute of the vector $\mathbf{x}$.

Additionally, an inverse matrix to a matrix of intervals can't be calculated for the reason of interval arithmetic limitations. Therefore the vector $\mathbf{w}$ from the linear regression (1) is calculated for midpoints of intervals which are k neighbours of the question point $\mathbf{x}^*$. Similarly, calculations of the optimal value of the w_N coefficient from the formula (3) are also performed for midpoints of chosen samples.

In experiments described in the section 4, for calculations on interval numbers there was applied Matlab toolbox INTLAB created by S.M. Rump and described in [17]. Mini-models created with the application of the interval arithmetic work correctly both for interval data and crisp data.

3.2 Information-Gap Model of Uncertainty

Data gathered by miscellaneous measurement equipment usually are burdened with a certain error. Such data can be stored in the form of interval numbers for which a possibility of mathematical calculations exists.

Other approach in modelling of uncertainty surrounding each data vector $\mathbf{x}_i$, $i = 1 \ldots M$, is representing it by defining a local information-gap model [4, 14]:

$$L(\alpha, \mathbf{x}_i) = \left\{\mathbf{x} \in \mathbf{R}^N :\ ||\mathbf{x} - \mathbf{x}_i||_\infty \leq \alpha\right\}, \quad \forall \alpha \in \mathbf{R} \tag{6}$$

where: $||\ldots||_\infty$ is the infinity norm in $\mathbf{R}^N$ defined as:

$$||\mathbf{x}||_\infty = \max_j |x_j|\,, \tag{7}$$

where: x_j is the j-th attribute of the vector $\mathbf{x}$ ($j = 1 \ldots N$) and $\alpha > 0$ is the unknown horizon of uncertainty. $L(\alpha, \mathbf{x}_i)$ can be treated as an unbounded family of hypercuboid sets (for infinity norm) of possible $\mathbf{x}_i$ realisations.

Let's assume that we have 2 normalised distinct data sets. First of them contains training data (although in the case of memory methods like the mini-models method there is no real training process) and each data sample consists of input vector $\mathbf{x}_k$ and target output value y_k, $k = 1 \ldots L$. The second data set contains testing data and each data sample also consists of input vector $\mathbf{x}_i$ and target output value y_i, $i = 1 \ldots M$.

For the crisp data, an error of modelling for the single testing sample $\mathbf{x}_i$ can be evaluated exemplary as:

$$\delta_i = |y_i - y_i^*| \,, \tag{8}$$

where: y_i^* is the model answer for the question point $\mathbf{x}_i$. A mean absolute error for the entire testing data set can be calculated as:

$$e_{\mathrm{MAE}} = \frac{1}{M} \sum_{i=1}^{M} |y_i - y_i^*| \,. \tag{9}$$

Now, we must take into account that data are uncertain so an accuracy of the model should be evaluated in a different way. The data are interval values so the accuracy will be also the interval value and its lower interval bound will be equal to a robustness function and an upper interval bound will be equal to an opportunity function [3]. Concepts of such functions are introduced by information-gap theory and are described later.

Both y_i and y_i^* will be interval numbers so their subtraction result can be evaluated as:

$$d_i = y_i - y_i^* = \left[\underline{y_i} - \overline{y_i^*}, \overline{y_i} - \underline{y_i^*}\right] ,$$

according to interval arithmetic [10]. The model error will be interval number:

$$\delta_i = [\underline{\delta_i}, \overline{\delta_i}] \,, \tag{10}$$

where the lower bound can be calculated as:

$$\underline{\delta_i} = \begin{cases} 0 & \text{if } y_i \cap y_i^* \neq \emptyset \\ \min\{|\underline{d_i}|, |\overline{d_i}|\} & \text{otherwise,} \end{cases} \tag{11}$$

and the upper bound as:

$$\overline{\delta_i} = \max\{|\underline{d_i}|, |\overline{d_i}|\} \,. \tag{12}$$

Fig. 4 illustrates the way of calculating lower and upper bound of the model error.

Now we can evaluate an interval value for the mean absolute error of the entire testing set as:

$$e_{\mathrm{MAE}} = \left[\underline{e_{\mathrm{MAE}}}, \overline{e_{\mathrm{MAE}}}\right] ,$$

where:

$$\underline{e_{\mathrm{MAE}}} = \frac{1}{M} \sum_{i=1}^{M} \underline{\delta_i} \quad \text{and} \quad \overline{e_{\mathrm{MAE}}} = \frac{1}{M} \sum_{i=1}^{M} \overline{\delta_i} \,.$$

Next, additionally let's define the notion of the model accuracy as:

$$q = \frac{1}{1 + e_{\mathrm{MAE}}} = [\underline{q}, \overline{q}] = \left[\frac{1}{1 + \overline{e_{\mathrm{MAE}}}}, \frac{1}{1 + \underline{e_{\mathrm{MAE}}}}\right] . \tag{13}$$

The accuracy defined in such way changes its value in the range from 0 (for model errors approaching infinity) to 1 (for model errors approaching 0). It is clear that always $\underline{q} \leq \overline{q}$.

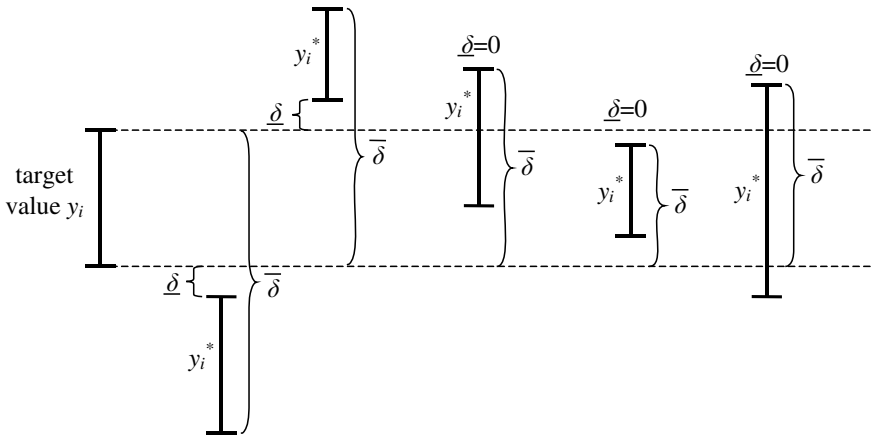


Fig. 4. Exemplary calculations of the lower and upper bound of the model error

Now, let's define the information-gap robustness function as equal to $\underline{q}$. Let r_c be the lowest acceptable model quality. The robustness of the model is the greatest horizon of uncertainty at which the model quality lower bound is equal or greater than r_c:

$$\hat{\alpha}(r_c) = \max\{\alpha : \ \underline{q} \geq r_c\} . \tag{14}$$

The $\underline{q}$ value decreases with increasing α (Fig. 6) so the robustness increases with decreasing r_c:

$$r_c < r_c' \quad \Rightarrow \quad \hat{\alpha}(r_c) \geq \hat{\alpha}(r_c') .$$

Next, we can define notions of the opportunity function equal to $\overline{q}$ and the opportuneness of the model as the lowest horizon of uncertainty at which the model quality upper bound is equal or greater than r_w value (windfall quality):

$$\hat{\beta}(r_w) = \min\{\alpha : \ \overline{q} \geq r_w\} . \tag{15}$$

$\hat{\beta}(r_w)$ is increasing as r_w is getting larger (Fig. 6):

$$r_w > r_w' \quad \Rightarrow \quad \hat{\beta}(r_w) \geq \hat{\beta}(r_w')$$

and it means that increasing the windfall quality of the model r_w causes an increase in the level of uncertainty $\hat{\beta}(r_w)$ needed to obtain that windfall.

4 Experiments

In the beginning, let's consider simple one-dimensional data presented in Fig. 5 (noisy sine function, 100 samples). The crisp data were submitted to the interval expansion of the size α (both input and output) in the way described in the subsection 3.1. Initially experiments proved that the interval input data are correctly propagated into the interval output by the mini-model approximator.

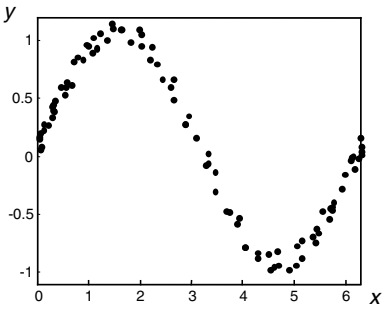


Fig. 5. Exemplary one-dimensional data applied in the experiment (before interval expansion)

In the next step, let's take under consideration the model based on the data from Fig. 5. The number of nearest neighbours k is set to 5, and the α value will grow from 0 to 0.1. We can evaluate the lower and the upper bound of the model quality from the equation (13) for linear (Fig. 6a) and nonlinear (Fig. 6b) mini-models.

Plots from Fig. 6 can be used to quantify the performance under uncertainty of a given model and directly access the robustness of the model at any demanded quality. For example, in the Fig. 6a (linear mini-models) we can see that by setting r_c to 0.9, an error of up to 0.025 can be tolerated in all input vectors $\mathbf{x}_i$, without the risk of decreasing quality below 0.9. That is, the robustness is $\hat{\alpha}(r_c) = 0.025$. Additionally, such level of uncertainty provides the opportunity for the model quality of up to 0.970, (opportunity function is $\hat{\beta}(r_w = 0.970) = 0.025$).

Respectively, values obtained for nonlinear mini-models can be evaluated from Fig. 6b. For the r_c equal to 0.9 the robustness is $\hat{\alpha}(r_c = 0.9) = 0.02$ and such level of uncertainty provides the opportunity for the model quality of up to 0.957. Realised experiments have shown that model based on linear mini-models is more robust to the data uncertainty than mini-model based on nonlinear ones.

The model robustness notion can be also applied for a discrimination between various models based on mini-models. For a certain level of uncertainty α, the best will be the model with the greatest value of $\hat{\alpha}$ and the lowest value of $\hat{\beta}$. Fig. 7 presents the plot of robustness function values for models with various nearest neighbour number k and the level of uncertainty α set to 0.03. Experiments were realised for linear mini-models and the data from Fig. 5. From the figure we can see that the most robust (for the uncertainty $\alpha = 0.03$) is the model with $k = 7$.

Of course, the described method for an evaluation of the model robustness can be also applied for a more complicated data with a greater number of input attributes. Exemplary calculations for the popular benchmark data 'cpu' are presented below. (The benchmark can be found in UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets.html.) Data have 6 input

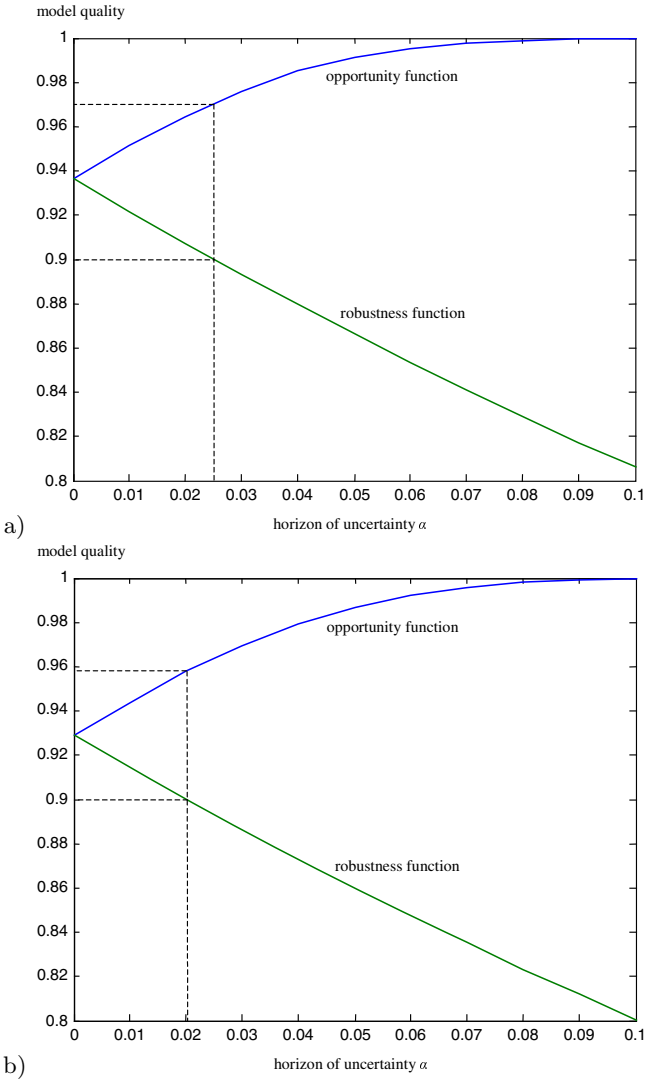


Fig. 6. The lower and the upper bound of the model quality for various α values for linear (a) and nonlinear (b) mini-models

attributes, so a normalisation was the first performed step. Next, the number k was set to 5 and α value was changed from 0 to 0.1. Fig. 8 presents the robustness function and the opportunity function for the 'cpu' data obtained for linear mini-models. If we set r_c to 0.9 we can find the model robustness $\hat{\alpha}(r_c = 0.9) = 0.013$. As before, it means that an error of up to 0.013 will not decrease quality below 0.9.

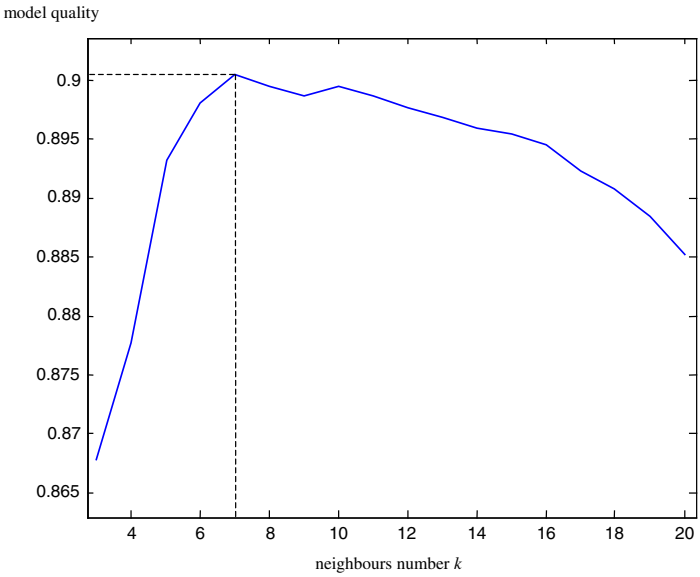


Fig. 7. The plot of robustness function values for models created with various neighbours number k and $\alpha = 0.03$

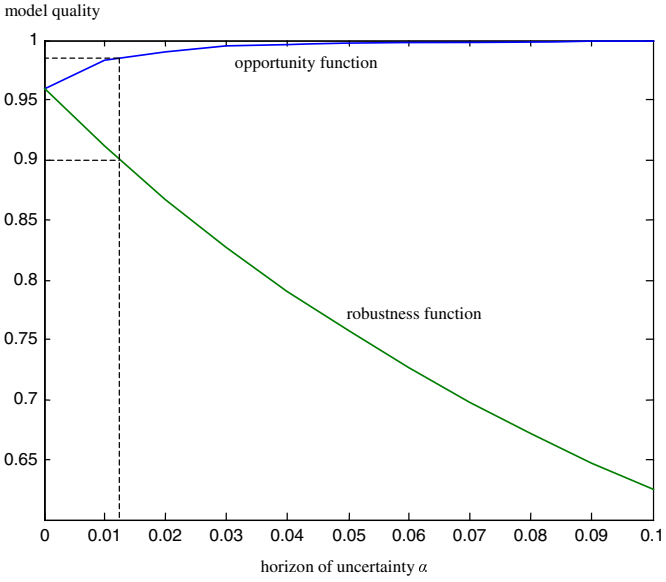


Fig. 8. The lower and the upper bound of the model quality for various α values (figure created for the 'cpu' data)

5 Conclusions

The paper has described an approach based on the information-gap theory which enables an evaluation of worst case error predictions of mini-models in the presence of a specified level of uncertainty in the data. There were presented concepts of the robustness and the opportunity of mini-models and calculations of these concepts were performed for the simple 1-D data set and next, for the more complicated 6-D data set. In both cases (and also in many other experiments realised by the author) the method worked correctly and enabled evaluation of the robustness and the opportunity for the given lowest acceptable quality r_c or the windfall quality r_w. The method enabled also choosing of the best (the most robust) model for the given level of uncertainty α.

References

1. Atkeson, C.G., Moore, A.W., Schaal, S.A.: Locally weighted learning. Artificial Intelligence Review 11, 11–73 (1997)
2. Ben-Haim, Y.: Set-models of information-gap uncertainty: axioms and an inference scheme. Journal of the Franklin Institute 336, 1093–1117 (1999)
3. Ben-Haim, Y.: Information-gap decision theory: decisions under severe uncertainty. Academic Press, New York (2001)
4. Ben-Haim, Y.: Uncertainty, probability and information-gaps. Reliability Engineering and System Safety 85, 249–266 (2004)
5. Cichosz, P.: Learning systems. WNT Publishing House, Warsaw (2000) (in polish)
6. Duch, W.: Uncertainty of data, fuzzy membership functions, and multilayer perceptrons. IEEE Transactions on Neural Networks 16(1), 10–23 (2005)
7. Hand, D., Mannila, H., Smyth, P.: Principles of data mining. The MIT Press (2001)
8. Kordos, M., Blachnik, M., Strzempa, D.: Do We Need Whatever More Than k-NN? In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part I. LNCS, vol. 6113, pp. 414–421. Springer, Heidelberg (2010)
9. Moore, A.W., Atkeson, C.G., Schaal, S.A.: Memory-based learning for control. Technical Report CMU-RI-TR-95-18, Carnegie-Mellon University, Robotics Institute (1995)
10. Moore, R.E., Kearfott, R.B., Cloud, M.J.: Introduction to interval analysis. Society for Industrial and Applied Mathematics, Philadelphia (2009)
11. Piegat, A.: Fuzzy modeling and control. Physica Verlag, Heidelberg (2001)
12. Piegat, A., Wąsikowska, B., Korzeń, M.: Application of the self-learning, 3-point mini-model for modeling of unemployment rate in Poland. Studia Informatica, University of Szczecin (2010) (in Polish)
13. Piegat, A., Wąsikowska, B., Korzeń, M.: Differences between the method of mini-models and the k-nearest neighbors on example of modeling of unemployment rate in Poland. In: Proceedings of 9th Conference on Information Systems in Management, pp. 34–43. WULS Press, Warsaw (2011)
14. Pierce, S.G., Ben-Haim, Y., Worden, K., Manson, G.: Evaluation of neural network robust reliability using information-gap theory. IEEE Transactions on Neural Networks 17(6), 1349–1361 (2006)

15. Plucinski, M.: Application of data with missing attributes in the probability RBF neural network learning and classification. In: Sołdek, J., Drobiazgiewicz, L. (eds.) Artificial Intelligence and Security in Computing Systems: 9th International Conference ACS 2002: Proceedings, pp. 63–72. Kluwer Academic Publishers, Boston (2003)
16. Pluciński, M.: Mini-models – local regression models for the function approximation learning. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 160–167. Springer, Heidelberg (2012)
17. Rump, S.M.: INTLAB – INTerval LABoratory. In: Developements in Reliable Computing, pp. 77–104. Kluwer Academic Publishing, Dordrecht (1999)
18. Wasserman, P.D.: Advanced methods in neural computing. Van Nostrand Reinhold, New York (1993)
19. Wright, W.A.: Bayesian approach to neural network modeling with input uncertainty. IEEE Transactions on Neural Networks 10(6), 1261–1270 (1999)

A Clustering Method Based on the Modified RS Validity Index

Artur Starczewski

Institute of Computational Intelligence, Częstochowa University of Technology
Al. Armii Krajowej 36, 42-200 Częstochowa, Poland
starcz@kik.pcz.czest.pl
http://kik.pcz.pl

Abstract. This paper describes a new method to the determination of the optimal number of well-separable clusters in data sets. The determination of this parameter is necessary for many clustering algorithms to define the naturally existing clusters correctly. In the presented method the idea of the agglomerative hierarchical clustering has been used, and the *modified RS* cluster validity index has been applied. In the first phase of the method, clusters are created due to the idea of hierarchical clustering. Then, for the optimal number of clusters the *k-means* algorithm is performed. The method has been used for multidimensional data, and the received results confirm very good performances of the proposed method.

1 Introduction

Clustering makes a split of the data elements into the homogeneous subsets (named clusters), inside which elements are more similar to each other, while they are more different in other groups. Nowadays, a large number of clustering algorithms exist and the algorithms can be divided into two basic groups, hard (crisp clustering) and fuzzy (soft clustering). Hard clustering algorithms make partitions of the data which are separable, i.e., each element of the set belongs to one cluster only. In lots of real data, the association of each element with clusters can have the fuzzy character, i.e., the membership to each cluster is partial and it creates the so called fuzzy partition. Clustering algorithms can also be classified into some categories, e.g., partitional clustering, hierarchical clustering, density-based clustering and other clustering. Partitional clustering algorithms create a one-level partitioning of the data, whereas hierarchical algorithms create multilevel partitioning. The popular and well-known algorithms of the partitional clustering are, e.g., *k-means* and its variations [2,4,8,12] or *nearest neighbour clustering* [10]. Among hierarchical methods one can mention: *single link*, *complete link* and *average link* [7,15]. Density-based clustering is capable of finding of arbitrary shaped clusters. In this approach, clusters are defined as dense regions separated by low density regions. A popular method, which represented this approach is called DBSCAN [3].

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 242–250, 2013.

The number of clusters, which should be created in a data set is a very important parameter. It influences correct performance of data clustering. For most algorithms, this parameter must be given *a priori*. Then, the so called *cluster validity index* is used, so the perfect partitions of a data set can be realized. Nowadays, lots of the indices to study the correctness of clustering data exist. Similarly to clustering algorithms, cluster validity indices are split into two groups: the indices for hard partitions and fuzzy partitions. The first group includes, e.g., *Dunn's index*, *SD validity index*, *RS index* [6], *S-index* [22], whereas to fuzzy partitions belong: *partition entropy index* [1], *validity based on fuzzy similarity* [14], *partition separation index* [24].

In this paper a new method to the determination of the optimal number of clusters in data sets is proposed. This method uses the *modified RS index*, which is named *mRS-index*. The paper is organized as follows. Section 2 describes the *RS* validity index and the *modified RS* index. Sections 3 presents the clustering method, which applies the new index. Section 4 illustrates the experimental results on synthetic data sets. Finally, there are conclusions in Section 5.

2 Modified RS Cluster Validity Index

Cluster validity indices are most often a ratio of intercluster and intracluster distance or vice versa. It can be also the sum of these distances. These distance make it possible to qualify two important properties of clusters, their separability and compactness. For the centre-base clusters, the Euclidean distance between their centres is the most often used separability measure. The compactness measure is often variance. Relatively to the type of the validity index, the best partition of a data set corresponds to the maximum or minimum index value. Let $\mathrm{C} = \{\mathrm{C}_1, \mathrm{C}, ..., \mathrm{C}_\mathrm{m}\}$ be a set of m-clusters, which creates a partition of the data set $\mathrm{X} = \{x_1, x_2, ..., x_n\}$. Instead, C_l, where $l = 1, .., m$ denotes a cluster in the partition of the data set and $\mathbf{v_l}$ is its prototype vector. Then the *RS* index [6,20] can be represented as below:

$$RS = \frac{SS_b}{SS_t} = \frac{SS_t - SS_w}{SS_t} \tag{1}$$

where SS_b is the sum of squares between clusters, SS_t is the total sum of squares of the whole date set and SS_w is the sum of squares within a cluster. The formula Eq.(1) can be expressed as follows:

$$RS = \frac{\sum_{j=1}^{d}\sum_{i=1}^{n_j}\left(x_{ji} - \widehat{v}_j\right)^2 - \sum_{l=1}^{m}\sum_{j=1}^{d}\sum_{i=1}^{n_{lj}}\left(x_{ji} - v_j\right)^2}{\sum_{j=1}^{d}\sum_{i=1}^{n_j}\left(x_{ji} - \widehat{v}_j\right)^2} \tag{2}$$

where d is a dimension of a vector $\mathbf{x}$ and $\widehat{\mathbf{v}}$ is the means of all the data points in set X and can be defined as:

$$\widehat{\mathbf{v}} = \frac{1}{|\mathrm{X}|}\sum_{\mathbf{x}\in X}\mathbf{x} \tag{3}$$

where |X| is the number of elements in the data set. In this paper for the optimal partition of a data set, the new $mRS - index$ is proposed, Eq.(7). The index contains the compactness measure of partition of the data set Eq.(6), where variance is used. Variance can be expressed as follows:

$$\sigma\left(C_l\right) = \frac{1}{|C_l|} \sum_{\mathbf{x} \in C_l} \left\|\mathbf{x} - \mathbf{v}_l\right\|^2 \tag{4}$$

where $|C_l|$ is the number of elements in the l-th cluster, and $\mathbf{v}_l$ is its centre (a prototype vector):

$$\mathbf{v}_l = \frac{1}{|C_l|} \sum_{x \in C_l} \mathbf{x}_l \tag{5}$$

Then, the clusters compactness of the partition of X can be defined as follows:

$$\sigma_{max} = \max_{1 \le l \le m} \left\{\sigma\left(C_l\right)\right\} \tag{6}$$

The new cluster validity index of the partition of the data set is expressed as follows:

$$V = RS - \alpha \frac{RS}{\sigma_{max}} \tag{7}$$

where the factor α is equal to the value of the clusters compactness Eq.(6), which is calculated for the initial 10-partition of a data set assigned by hierarchical clustering algorithm *single link*, that is α equals $\sigma_{max}^{(10)}$. The value of the factor has been defined experimentally and has been correct for different data sets. The formula Eq.(7) for k-partitions of a data set can be expressed as follows:

$$V_{(k)} = RS_{(k)} - \alpha \frac{RS_{(k)}}{\sigma_{max}^{(k)}} \tag{8}$$

This additional component can be distinguished as follows:

$$RC_{(k)} = \alpha \frac{RS_{(k)}}{\sigma_{max}^{(k)}} \tag{9}$$

This component has a strong influence on getting a correct value of the validity index, which assigns optimal partition of a data set. Note, using a hierarchical algorithm for clustering, for a number of clusters $m > o$, where o is the optimal number of clusters, the value of clusters compactness Eq.(6) is not large enough. When $m - 1$, it can increase in a low range. For a number of cluster equal to optimal $m = o$, the compactness does not increase sharply too. That is so, because the clusters are still quite compact, so the variance value is not large. However, when $m < o$, then the value Eq.(6) increases sharply, so the clusters compactness is low. The first high increase of the clusters compactness appears when the number of clusters is $m = o-1$. Then the value of additional component Eq.(9) is decreased and the validity index Eq.(8) has a maximal value. In order to get the number of clusters equal to o the value m has to be increased $m + 1$. The modification of RS validity index by this additional component Eq.(9) greatly improves assignation of the optimal partitions of a data set. In the next section, a method which uses the novel cluster validity index is proposed.

3 A Clustering Method Based on the Modified RS Validity Index

This proposed clustering method employs *single link* agglomerative hierarchical algorithm. The method can be described in the following steps.

1. Start the new clustering method:
   (a) merger clusters in a data set according to the idea of the hierarchical algorithm *single link* starting with $m = n$.
   (b) for each k-partition calculation of cluster validity index using the formula Eq.(8).
   (c) recording values of cluster validity index in the table $R[m]$.
   (d) stop for $m = 1$.
2. Assignation of the optimal number of clusters:
   (a) finding the index m for the maximum value of the table $R[m]$,
   (b) increasing the index $m = m + 1$,which is equal to the optimal number of clusters in the data set.
3. Start of algorithm *k-means* for the optimal number of clusters $o = m$.

Below the testes results realized for a various number of clusters are presented.

4 Experimental Results

The machine learning toolkit Weka 3.6 [23] has been used to generate data sets. The first two generated data sets are 2-dimensional with a different number of clusters and the next sets are multidimensional: 20-dimensional and 30-dimensional. The real data sets, i.e., the Iris and the Wine are from UCI machine learning repository [11].The sets have been clustered by means of the new method described in Section 3. The clustering method does not require introducing the parameter determining the number of clusters. It is calculated during the performance of the algorithm. Because the number of cluster is a key parameter for the correct data partition, the clustering results presented in the Table 2 and Table 3 include only this parameter. The appropriate number of clusters indicated by the parameter, due to naturally existing groups, can ensure also the correct qualification of the element membership of the sets to the clusters.

4.1 2-Dimensional Data on Different Number of Clusters

In the Fig. 1 two synthetic 2-dimensional sets are presented. The detailed description of those sets is described in Table 1. The data set A consists of 4 clusters, while the set B contains 8 clusters. The number of elements in clusters is different. Clusters are located at various distances from each other, some of them are quite close. The clustering is performed by means of the method described in Section 3. The values of the *RS-index* (see Eq.(1)) and *mRS-index*

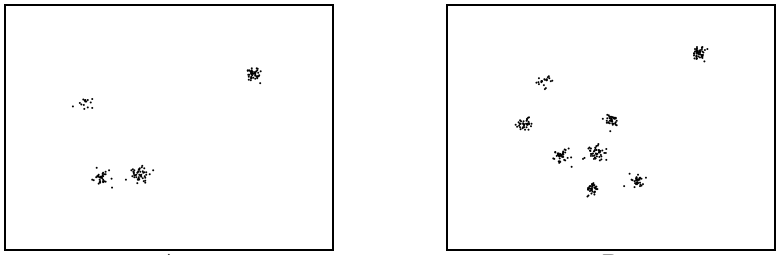
Table 1. Detailed information of the data sets

Data set	Clusters	Number of attributes	Total number of instances
A	4	2	146
B	8	2	300
C	5	20	102
D	12	20	211
E	4	30	50
F	10	30	164
Iris	3	4	150
Wine	3	13	178

(see Eq.(8)) obtained for different partitions of A and B data sets are presented in Fig. 2. In those figures the axis x means the number of clusters, while the axis y is the value of the validity index. We can see that all the time the *RS* index values increase slightly, the determination of the optimal partition of the data set is not easy. Instead, when the *modified RS* index is used the maximum value is exactly defined. Note, that the *mRS-index* assigns the optimal partition of data A and B (that is 4 and 8), see Tabele 2. The table contains only the number of obtained clusters, as it has been mentioned earlier, this is the key parameter to perform data clustering correctly.

Table 2. Number of clusters determined by means of *mRS-index*

Data sets and optimal number of clusters		*mRS-index* number of clusters
A	4	4
B	8	8



A B

Fig. 1. Graphic presentation of A and B data sets

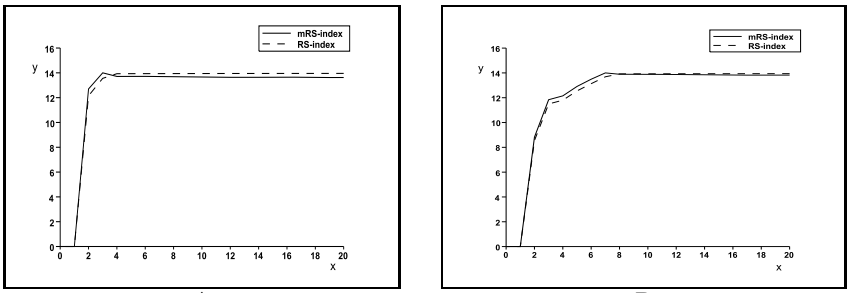


Fig. 2. Values of the *mRS* and *RS* index depending on the number of clusters in A and B data sets

4.2 Multi-dimensional Data on Different Number of Clusters

Similarly to Section 4.1, the new method has been used to clustering the multi-dimensional data sets. In Table 1 the sets are presented. Some obtained values of the validity index, Eq.(1) and Eq.(8), are depicted in Fig. 3 and Fig. 4.

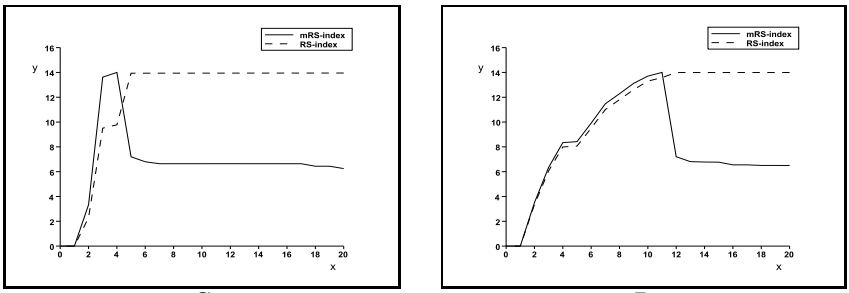


Fig. 3. Values of the *mRS* and *RS* index depending on the number of clusters in C and D data sets

The number of clusters obtained by means of the *mRS-index* has been presented in Table 3. It can be seen, that despite the large number of clusters and high dimensionality of sets, the new clustering method defined in Section 3 has assigned the optimal number of clusters in sets correctly. The performed experiments confirm the high effectiveness of the method.

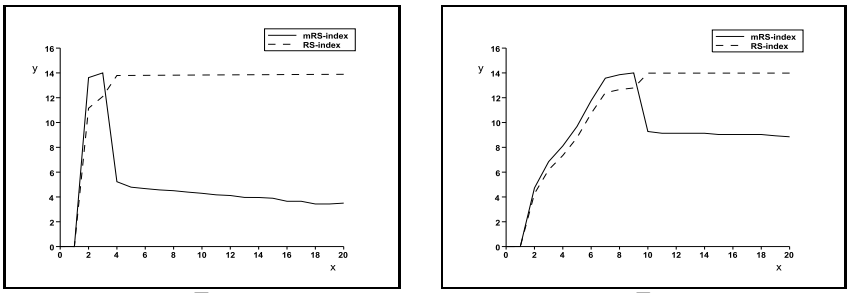


Fig. 4. Values of the *mRS* and *RS* index depending on the number of clusters in E and F data sets

Table 3. Number of clusters determined by means of *mRS-index*

Data sets and optimal number of clusters		*mRS-index* number of clusters
C	5	5
D	12	12
E	4	4
F	10	10
Iris	3	3
Wine	3	3

5 Conclusions

All the presented results confirm the very good effectiveness of the new clustering method (Section 3), where the number of clusters is assigned correctly for the clusters of different size and distance from each other. In this method, the *modified RS* validity index is applied, which plays key rule for the correct partition of data sets. The hierarchical algorithm used in the method also has influence on determination of the optimal number of clusters in sets. In the described method, *single link* algorithm has been used, because of its popularity and simplicity in practical implementation. Other versions of hierarchical algorithms, e.g., *complete* or *average link* have other properties and they can have influence on values of the new cluster validity index, Eq.(7), but it requires further studies. When the test sets includes data noise, additional methods which remove it have to be applied. To sum up, the suggested method requires some further research, but the received results are already very promising. The ongoing research is focused on applications of obtained results to create various neuro-fuzzy systems [9,17,21] and design probabilistic neural networks working both in stationary [5,16] and non-stationary environment [18,19].

References

1. Bezdek, J.C.: Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York (1981)
2. Bradley, P., Fayyad, U.: Refining initial points for k-means clustering. In: Proceedings of the Fifteenth International Conference on Knowledge Discovery and Data Mining, pp. 9–15. AAAI Press, New York (1998)
3. Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc 2nd KDD. AAAI Press (1996)
4. Faber, V.: Clustering and the continuous k-means algorithm. Los Alamos Science 22, 138–144 (1994)
5. Greblicki, W., Rutkowski, L.: Density-free Bayes risk consistency of nonparametric pattern recognition procedures. Proceedings of the IEEE 69(4), 482–483 (1981)
6. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Clustering validity checking methods: Part II. ACM SIGMOD Record 31(3) (2002)
7. Jain, A., Dubes, R.: Algorithms for clustering data. Prentice-Hall, Englewood Cliffs (1988)
8. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Comput. Surveys 31(3), 264–323 (1999)
9. Li, X., Er, M.J., Lim, B.S., et al.: Fuzzy Regression Modeling for Tool Performance Prediction and Degradation Detection. International Journal of Neural Systems 20(5), 405–419 (2010)
10. Lu, S.Y., Fu, K.S.: A sentence-to-sentence clustering procedure for pattern analysis. IEEE Trans. Syst. Man Cybern. 8, 381–389 (1978)
11. Mertez, C.J., Murphy, P.M.: UCI repository of machine learning databases, `http://www.ics.uci.edu/pub/machine-learning-databases`
12. Mcqueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistic and Probability, pp. 281–297 (1967)
13. Murtagh, F.: A survey of recent advantces in hierarchical clustering algorithms. The Computer Journal 26(4), 354–359 (1983)
14. Pei, J., Yang, X.: Study of clustering validity based on fuzzy similarity. In: Proceedings of the 3rd World Congress on Intelligent Control and Automation, Hefei, China, pp. 2444–2447. IEEE (2000)
15. Rohlf, F.: Single link clustering algorithms. In: Krishnaiah, P., Kanal, L. (eds.) Handbook of Statistics, vol. 2, pp. 267–284. North-Holland, Amsterdam (1982)
16. Rutkowski, L.: A general approach for nonparametric fitting of functions and their derivatives with applications to linear circuits identification. IEEE Transactions Circuits Systems CAS-33, 812–818 (1986)
17. Rutkowski, L., Przybył, A., Cpałka, K.: Novel Online Speed Profile Generation for Industrial Machine Tool Based on Flexible Neuro-Fuzzy Approximation. IEEE Transactions on Industrial Electronics 59(2), 1238–1247 (2012)
18. Rutkowski, L.: Generalized regression neural networks in time-varying environment. IEEE Trans. Neural Networks 15, 576–596 (2004)
19. Rutkowski, L.: Adaptive probabilistic neural-networks for pattern classification in time-varying environment. IEEE Trans. Neural Networks 15, 811–827 (2004)
20. Sharma, S.: Applied Multivariate Techniques. John Wiley & Sons, New York (1996)

21. Starczewski, A.: A new approach to creating multisegment fuzzy systems. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 324–332. Springer, Heidelberg (2008)
22. Starczewski, A.: A cluster validity index for hard clustering. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 168–174. Springer, Heidelberg (2012)
23. Weka 3: Data Mining Software in Java,University of Waikato, New Zealand, `http://www.cs.waikato.ac.nz/ml/weka`
24. Yang, M., Wu, K.: A new validity index for fuzzy clustering. In: Proceedings of the 10th IEEE International Conference on Fuzzy Systems, vol. 1, pp. 89–92 (2001)

Online Emotion Classification from Electroencephalographic Signals: A First Study Conducted in a Realistic Movie Theater

Thierry Castermans, Matthieu Duvinage, and Nicolas Riche

University of Mons, Faculty of Engineering, TCTS Lab.,
Place du Parc, 20, 7000 Mons, Belgium

Abstract. New affordable electroencephalographic (EEG) devices dedicated to the game industry have recently appeared on the market. Consequently, more and more artists inspired by these promising technologies imagine the possibility to directly integrate human emotions in their performance through measurement of the brain waves. In this paper, we describe a first attempt to detect the valence (positive or negative value) of emotions provoked by different video excerpts amongst several spectators seated in a realistic movie theater, on the basis of their EEG signals only. Preliminary results obtained with consumer grade EEG headsets indicate that positively and negatively valenced video excerpts may be discriminated in the training data set but not on an independent data set. These results are extensively discussed as well as further studies and developments that should be made to improve performance in the future.

Keywords: Electroencephalography, emotion, video.

1 Introduction

Since the first recording of an electroencephalogram by Hans Berger in 1924, electroencephalography (EEG) has become an essential tool used by physicians to detect different neural pathologies and – in conjunction with other techniques – to study sleep disorders. More recently, EEG signals have been exploited to develop Brain-Computer Interfaces (BCI). This kind of interface allows disabled people to communicate [2], control computers [7] and drive robotic [3] or prosthetic devices [4] via the power of their brain only. In short, BCI are developed according to the following scheme: different intentions or mental events have to be elicited in the brain of the user. Brain activity is then recorded (generally using EEG, which is a non-invasive technology); signals are cleaned, in order to get rid of the different artifacts (eye movements, muscle activities, ...) polluting the raw data; relevant features are then extracted and injected in a pattern recognition algorithm. Finally, according to the classification results, pre-defined commands are sent to the computer or the device to be controlled. In practice, one may activate high level commands like "go straightforward", "turn left", "turn right" in the case of a motorized wheelchair or different keyboard letters

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 251–262, 2013.

in the case of a text-speller application. The user is then provided with a feedback from the computer or the device, in order to understand which command or action was detected by the system.

With the recent advent of affordable electroencephalographic devices on the market, Brain-Computer Interfaces are becoming available to a broad public. These consumer grade headsets are primarily dedicated to provide a new control in video games but other applications can be imagined. Artists, for instance, are definitely willing to use these cheap headsets in order to integrate brain waves in their performances. In this context, we present a preliminary study to evaluate the possibility to integrate brain waves in the control of special effects added to a movie, a concept called "Emotive Cinema" in the following of this paper. Section 2 describes the objectives of this paper and the technical challenges encountered. The evaluation of the quality of the Emotiv EPOC headset (`http://www.emotiv.com`) and the resolution of different technical difficulties are given in section 3. The experimental approach and preliminary results related to emotion recognition using EEG signals are presented in section 4. Our "Emotive Cinema" setup is described in section 5. A discussion of this study and future work that could be envisaged is then given before final conclusions.

2 Objectives and Technical Challenges

This research project was set up on the basis of a specific demand from an artist team desiring to transpose the gamebook literary genre to the world of cinema. Ultimately, the idea is to influence the story of a film taking into account the emotional state of the audience. In a first step, the objective consists in controlling special cinema effects using affective information coming from the audience. The realization of such interface raises multiple questions and requires practical problems to be solved:

1. Is it possible to extract information related to the affective state of the audience analyzing the EEG signals of the spectators, in particular with low cost EEG headsets ?
2. Is it possible to connect and synchronize several headsets and thereby develop a technical solution to experiment further ?
3. To what extent large amounts of data can be sent to the computer ? How many headsets can be connected to the same computer ?
4. Movements of people can be captured by the EEG headsets, through the electrical activity of the muscles. Is this considered as problematic in the detection of the affective state of the spectators ? Does one have to integrate artifact removal techniques to limit the interferences ?

The originality and interest of this paper reside in the fact that a majority of the above questions are addressed and, additionally, that the study was conducted online in a realistic movie theater.

3 The Emotiv EPOC Headset

3.1 Hardware Quality Assessment

In a previous work, we had already assessed the quality of Emotiv EPOC headsets, by comparing their performance in the framework of a P300-based brain-computer interface with the performance reached using a medical EEG system (on the basis of the same electrode configuration, of course) [1]. Fig. 1 illustrates that, albeit giving worse results than those obtained with the medical system, Emotiv EPOC headsets are usable for this kind of BCI applications.

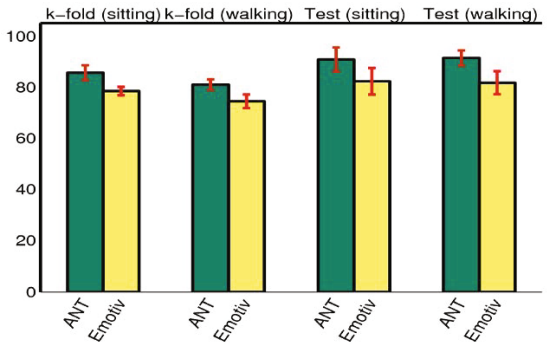


Fig. 1. This Figure illustrates that Emotiv headsets are usable for a P300-based BCI (chance level is 25 % - Figure adapted from [1])

3.2 Interconnectivity Issue

The framework chosen to develop the "Emotive Cinema" experimental setup is Openvibe (`http://openvibe.inria.fr`), an open source software dedicated to Brain-Computer Interfaces and real time neurosciences. Indeed, to our knowledge, this is the only software able to manage the synchronization of several acquisition systems (i.e. several headsets in our case). In practice, we found that Openvibe did not correctly receive the signals from two or more Emotiv headsets. In fact, signals from Emotiv headsets are transmitted to the computer without any wire. USB dongles (one per headset) receive the encrypted raw data that must subsequently be decoded using the licensed Emotiv SDK software. Actually, the signals obtained in Openvibe were completely messed up, as if Openvibe tried to listen to all USB dongles at the same time. We found that the Openvibe driver for the Emotiv EPOC headset was correctly written on the basis of the documentation given with the Emotiv Research Edition SDK_v1.0.0.4-PREMIUM. We solved the problem by using the Emotiv Research Edition SDK_v1.0.0.5-PREMIUM and by modifying the Openvibe driver in order to make sure that the origin of each data packet was correctly identified (this correction has now been integrated in the last official version of Openvibe).

4 Emotion Recognition Using EEG

4.1 Emotion Modelling and Self-Assessment Manikin (SAM)

In psychology, one typical way to characterize human emotions consists in projecting them into a 3-dimensional space [5]. The first axis of this space is *valence*, which represents the positive, neutral or negative aspect of the emotion. *Intensity* of the emotion is evaluated on the second axis. It represents the arousal felt during a given stimulus. Finally, *dominance* indicates to what extent the emotion felt was controllable or not. For instance, breaking into tears or hard laughing are examples of uncontrollable emotions, characterized by a small dominance value. In certain cases, limiting the decomposition to 2 dimensions (intensity and valence) is considered as sufficient. In this particular way of characterizing emotions, *well-being* would be represented by a positive valence and a small intensity value, *joy* by positive valence and high intensity, *sadness* by a negative valence and small intensity and *anger* by a negative valence and high intensity.

As emotions are personal feelings that can not be accurately measured by specific apparatus, the standard way to determine them consists in asking subjects to fill in a questionnaire. The Self-Assessment Manikin (SAM) questionnaire [6] is a non-verbal pictorial assessment technique which divides the valence, intensity and dominance axes in five intervals each, as represented on Fig. 2. The SAM ranges from a smiling, happy figure to a frowning, unhappy figure when representing the pleasure dimension, and ranges from an excited, wide-eyed figure to a relaxed, sleepy figure for the arousal dimension. The dominance dimension represents changes in control with changes in the size of SAM: a large figure indicates maximum control in the situation. During an experiment, each subject is asked to put an 'x' on the three axes (each divided in 5 intervals) in order to describe as best as possible the emotion he/she felt in front of a given stimulus. The results can be converted to numerical values afterwards.

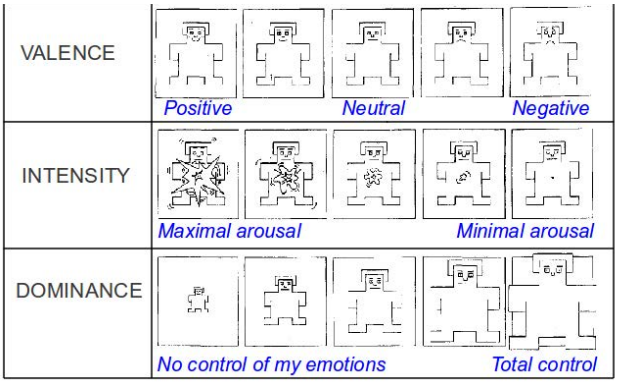


Fig. 2. Each subject describes the emotion felt in front of a given stimulus by putting an 'x' on each axis of the five scale SAM questionnaire

4.2 Experimental Approach

Protocol. The experiment consisted in presenting five video excerpts of 2 or 3 minutes each to three subjects wearing an Emotiv EPOC headset. The objective was to provoke a large variety of emotions by the subjects, by successively presenting them quiet landscapes of idyllic islands, a depressive man, a war scene, an excerpt of a one-man show of a french humorist and finally the recording of an impressive electronic cellos duo. After watching each short movie, the three participants were asked to fill in a SAM questionnaire. The average results of the three questionnaires are shown on Fig. 3. One can observe that all the regions of the intensity-valence plane are sampled by the 5 chosen video excerpts. EEG signals were stored on disk for offline analysis.

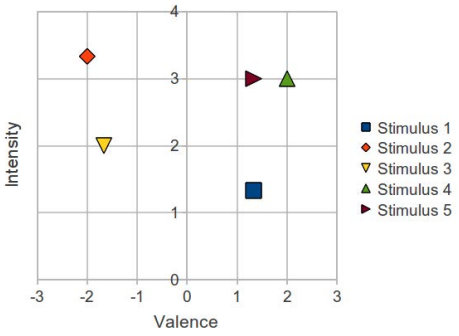


Fig. 3. The average SAM results indicate that all the regions of the intensity-valence plane are sampled by the 5 chosen video excerpts found on the youtube website

Asymmetry in Frontal Brain Activity and Emotion Valence. Our analysis is based on the results published by Schmidt *et al.* [6] who showed that:

"...positively valenced musical excerpts elicited greater relative left frontal EEG activity, whereas negatively valenced musical excerpts elicited greater relative right frontal EEG activity ...In addition, positively valenced (i.e., joy and happy) musical excerpts elicited significantly less frontal EEG power (i.e., more activity) compared with negatively valenced (i.e., fear and sad) musical excerpts, and there was significantly less frontal EEG power (i.e., more activity) in the left compared with the right hemisphere across valence."

In other words (cf. Fig. 4), measuring the asymmetry of the frontal brain activity would enable to discriminate the valence (+ or -) of emotions, at least from the musical point of view. Several other publications in emotion recognition using electroencephalography give similar results. In very brief outline, the analysis scheme is generally about the same in these studies: 1) compute the power of EEG in different frequency bands, 2) combine these values in some way, 3) use different classifiers and 4) evaluate the results.

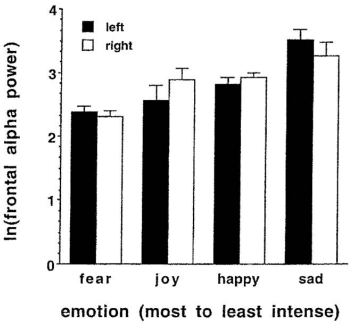


Fig. 4. Differences among four musical excerpts on left and right frontal EEG alpha power. Error bars represent the standard error of the mean (Figure from [6]).

Signal Processing. We decided to select frontal electrodes of the Emotiv headset, according to the prescription given in [6]. To compare the activity of the left and right brain hemispheres, we computed the difference between the EEG signals measured by symetric electrodes, namely AF3-AF4, F3-F4, F7-F8, FC5-FC6, as defined in the international 10/20 system. Signals were then band-pass filtered with a 4th order Butterworth filter whose low and high cut frequencies were respectively set to 8 and 12 Hz for the alpha band, 12 and 30 Hz for the beta band, 30 and 60 Hz for the gamma band. Note that given EEG signals were acquired using an apparatus running on a battery, no 50 Hz or 60 Hz line noise had to be removed. The sampling frequency of the Emotiv headset is fixed to 128 Hz. No additional artifact cleaning algorithm was used.

EEG data were divided in overlapping windows of 4 seconds duration every 0.5 second. Thus, two times per second, 12 features (i.e. 3 powers alpha, beta and gamma for the 4 frontal electrode differences) were sent to a simple Linear Discriminant Analysis (LDA) classifier. This classifier was trained using one minute of stimulus S1 (the movie with idyllic islands) in order to learn the EEG characteristics of a positively valenced emotion and one minute of stimulus S2 (the movie with the depressive man) in order to learn a negatively valenced emotion. The Openvibe software was used to realize this signal processing and classification task. Training was done using a 12-fold procedure. A k-fold test generally allows better classification rates. The idea is to divide the set of feature vectors in a number of partitions. The classification algorithm is trained on some of the partitions and its accuracy is tested on the others. The classifier with the best results is then selected as the trained classifier.

4.3 Preliminary Results

The performances of the LDA classifier after the training procedure are presented in Table 1. Good results demonstrate the possibility to discriminate EEG signals related to stimuli S1 and S2 (in the training data set), on the basis of the signal processing described in previous section. However, these encouraging

Table 1. Performance of the LDA classifier for the training and test data sets obtained for each subject

Subject	12-fold (%)	Sigma (%)	Test (%)
1	94	4	53
2	99	1	49
3	91	6	51

results must be correctly validated with a test dataset before making any valid conclusion. We thus computed the output of the trained LDA classifier when it was fed with EEG data (processed as described above) recorded during the 5 different movies (i.e. EEG data not exploited during the training phase). Fig. 5 shows the LDA classifier results as a function of time for one representative subject. In this plot, the blue points indicate the valence result given by the LDA classifier. Positive (negative) valence results are depicted by the value +1 (-1). With a perfect classifier, we should always get +1 values for positively valenced movies (i.e. stimuli 1, 4 and 5) and -1 values for negatively valenced movies (i.e. for stimuli 2 and 3). In practice, one observes that this is not the case. On average, the classifier gives the correct answer only about half of the time. This is actually the chance level for a two-state decision. Notice however that the LDA gives the correct answer during the entire stimulus 1.

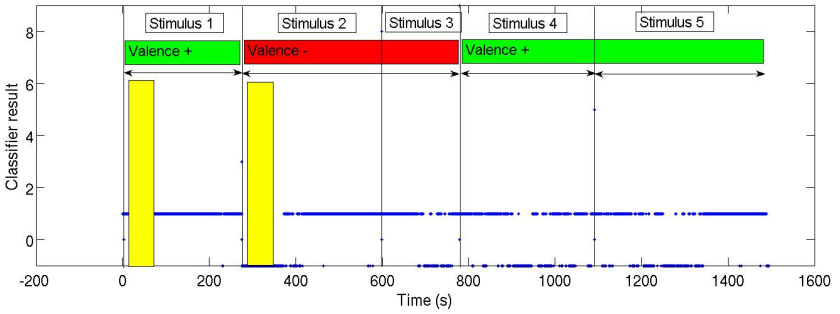


Fig. 5. LDA classifier results as a function of time for one representative subject. Positive and negative valence results are respectively depicted by +1 and -1 blue points. Stimuli 1, 4 and 5 were positively valenced whereas stimuli 2 and 3 were negatively valenced. Yellow rectangles indicate the training data set.

5 Emotive Cinema Setup

5.1 General Scheme of the Setup

In our experimental "Emotive Cinema" setup, three Emotiv headsets are sending raw EEG data to a Windows PC, via three corresponding USB dongles. Acquired raw data are processed in real time in the Openvibe software

following the pipeline described in section 4.2. Openvibe communicates the LDA decisions via a Virtual Reality Peripheral Network (VRPN) client/server (`http://www.cs.unc.edu/Research/vrpn/`) in which we have incorporated Oscpack (`http://www.rossbencina.com/code/oscpack`) functionalities in order to send OSC packets through the local network. These packets are received by an Apple machine running MaxMSP software, which plays the cinema with added special effects. In practice, the Openvibe scenario we used is shown on Fig. 7 (this represents the data processing for one spectator only).

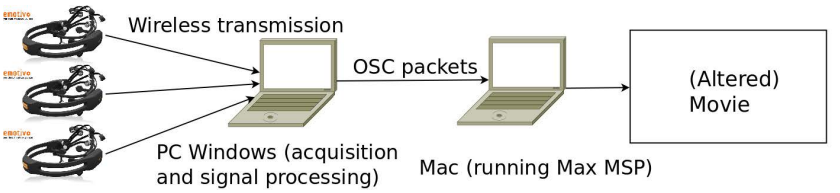


Fig. 6. Our experimental "Emotive Cinema" setup. A windows machine collects raw data from Emotiv EPOC headsets and processes them through the Openvibe software. Openvibe communicates the LDA decisions via the VRPN client/server which can send OSC packets through the local network. These packets are received by an Apple machine running MaxMSP software, which plays the cinema with added special effects.

5.2 Parameters Used to Control Cinema Effects

The philosophy adopted in this research consisted in the detection of valence (+ or -) of emotion felt by spectators watching different movie excerpts in order to control diverse cinema effects. This is a simple 2-class detection problem for which we used an LDA classifier. Mathematically, the LDA gives, once trained, the position of an hyperplane which optimizes the classification performance. The distance separating each sample from that hyperplane is computed and the sign of this value indicates to which class the sample belongs. Instead of using this categorical information to control cinema effects, we decided to use the distance to hyperplane in order to control cinema effects. Several reasons support this decision: (i) this is a more refined information than a simple categorical value. Indeed, small distances (in absolute values) indicate that the samples are close to the boundary. Hence, their distinction is less clearly established than for samples far away from the hyperplane. Using the distance allows in some way to introduce the notion of confidence in the classification. (ii) The distance is fluctuating on a certain range of values, instead of the discrete [-1,+1] interval. It thus gives many more possibilities to modulate and control cinema effects. (iii) We think that these values are intrinsically more normalized across people than EEG band power values. It is thus more legitimate to use these values in the case where a comparison between different spectators is desired.

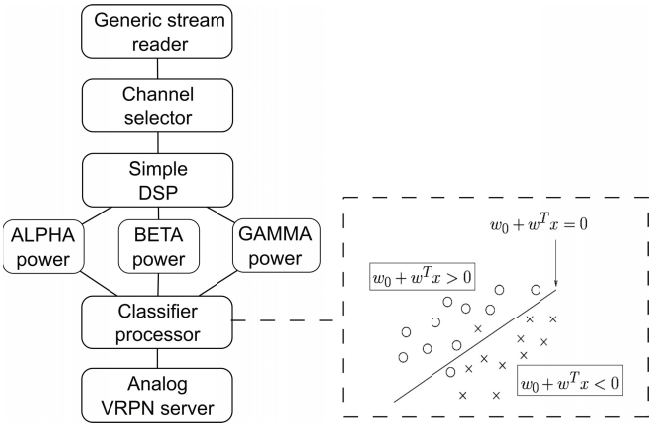


Fig. 7. In our Openvibe scenario, the difference between symetric electrodes of the frontal region is computed. Alpha, beta and gamma powers are calculated in windows of 4 seconds duration overlapping every 0.5 second. A simple LDA classifier is trained to discriminate positively and negatively valenced emotions. The distance to the LDA hyperplane – which separates at best the two classes samples – is sent under OSC packets through the local network via an adapted VRPN server.

5.3 Max MSP Interface

In our experimental "Emotive Cinema" setup, the control of simple cinema effects was carried out with a Max MSP interface (`http://cycling74.com`). Max is a platform for real-time signal processing, which allows both fast prototyping by using visual programming with libraries supported by a large community and flexibility by the possibility to build additional blocks if needed.

The Max MSP interface collects the OSC packets sent by the VRPN server. These packets contain, for each Emotiv EPOC headset, the alpha, beta and gamma band powers as well as an estimation of the valence of the emotion (i.e. the distance to LDA hyperplane) felt by the user. All these values can be plotted in real time for visual inspection (cf. Fig. 8).

As a very simple cinema effect, we used the value of the valence to modify the brightness of the video (the more positive the valence, the brighter the image). The interface allows the estimation of the valence synchronized with the video to be saved on disk as a Max MSP Binary File. This option makes it possible to build a database useful for long term analyses.

Fig. 9 shows an effect obtained with three users. Frontal brain activity of each user influences the brightness of one part of the screen. Of course, these simple illustrations are only given as basic examples of "artistic" possibilities. Other feedbacks are possible and it is up to the film makers to develop this aspect. For instance, the valence values of all the users could be compared. If small (large) discrepancies are found, a harmonious (dissonant) sound could be produced, as auditory feedback. Another effect could also link the light intensity of the room with the estimation of the valence.

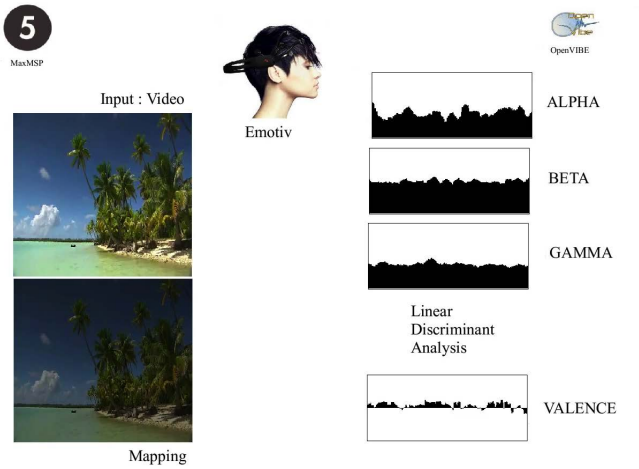


Fig. 8. Our Max MSP interface: on the right are plotted the alpha, beta and gamma EEG band powers (one user) with an estimation of the valence of his/her emotion. On the left, both the original and altered movies are shown. In this case, brightness of the movie is modified according to the valence estimated value (simple linear mapping).

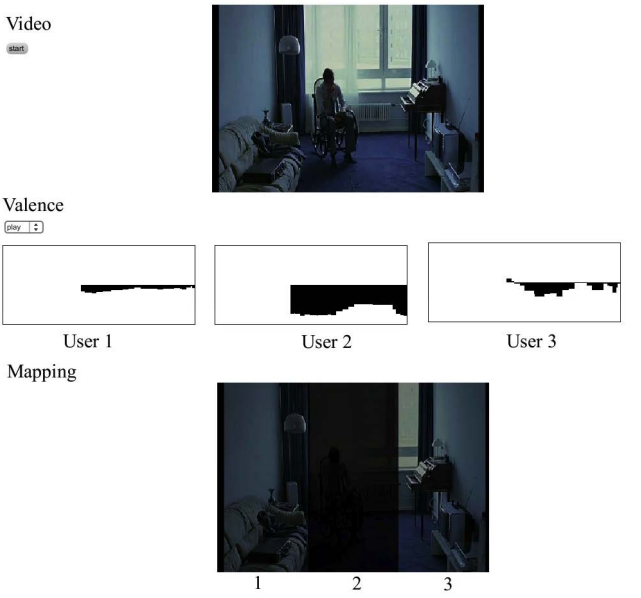


Fig. 9. Frontal activity of each user influences the brightness of one part of the screen

6 Discussion and Future Work

The preliminary results obtained are encouraging (excellent classification results in the training phase) and frustrating (chance level results in the testing phase) at the same time. In reality, the LDA classifier has a tendency to overfit when the ratio of the training samples to dimensionality is low. Increase this ratio seems necessary in this case in order to improve the performance of the classifier. Also, the question arises of the appropriateness of our experimental protocol. Indeed, is it legitimate to use a SAM questionnaire to characterize one global emotion felt during a movie of several minutes duration ? It seems reasonable to think that emotions will fluctuate on such timescales. This hypothesis is in agreement with the feedback given by the subjects who participated in this study. Considering this, the way we are computing the classifier performance in the test dataset becomes highly questionable. After all, one may even think that it is not possible at all to check the validity of the results without a device or sensor able to determine precisely (and on a short timescale) the emotional state of the user. Unluckily, to our knowledge, no such device is available on the market today.

Nevertheless, other technical points should be checked in a future work:

1. Instead of using only frontal electrodes, could an elaborated spatial filter (e.g. Common Spatial Pattern - CSP) help in this task ?
2. Would it be beneficial to combine two LDA classifiers, i.e. one dedicated to the discrimination of valence and the other one dedicated to the intensity of emotions ?
3. We trained the classifier using stimuli S1 and S2, which are differing on both axes of the valence-intensity plane. Is this problematic ? Would it be better to use two stimuli varying only in valence (and not in intensity) in the training data set ?
4. Is there a better strategy than the one described in Schmidt *et al.* [6] and on which we based our investigations ?

Other approaches, fundamentally different, could also be envisaged for the development of future "emotive cinema" applications. For instance, a classifier could be trained to recognize a relaxed/focused state, asking the subjects to intentionally place themselves in such emotional states.

7 Conclusions

Nowadays, emotion recognition using electroencephalography is a hot research topic. Many publications are being produced on the subject, with the aim of developing new human machine interfaces. With the advent of cheap EEG headsets on the market, the number of artists desiring to integrate brain waves in their performances has dramatically increased. This research project was set up on the basis of a specific demand from a team of film makers willing to add

and control special cinema effects using affective information coming from the audience.

In this project, we have developed and provided a functional solution of "Emotive Cinema", incorporated in a competitive and open-source BCI development framework. We have shown that at least three Emotiv headsets could be used to control in real time simple cinema effects. The control of these effects is done using an estimation of the valence of the emotions felt by three different spectators. Our preliminary results suggest that the robustness of this estimation is weak but this conclusion should be considered as preliminary. Indeed, the intrinsic complexity of measuring the evolution of various emotions felt by spectators on a several minutes duration video excerpt complicates the interpretation of the results. This demonstrates the need for further developments in this exciting research area. Artistic performance can definitely be realized using such biosignals provided no shortcut is taken in the interpretation of emotions with the experimental setup presented here.

Acknowledgments. This research was conducted with the Numediart institute (www.numediart.org) directed by Pr. Thierry Dutoit. M. Duvinage is a FNRS (Fonds National de la Recherche Scientifique) Research Fellow. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its author(s).

References

1. Duvinage, M., Castermans, T., Petieau, M., Hoellinger, T., De Saedeleer, C., Seetharaman, K., Cheron, G., Dutoit, T.: A P300-based quantitative comparison between the emotiv epoc headset and a medical EEG device. In: Proceedings of the 9th IASTED International Conference on Biomedical Engineering (2012)
2. Farwell, L.A., Donchin, E.: Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalography and Clinical Neurophysiology 70(6), 510–523 (1988)
3. Leeb, R., Friedman, D., Müller-Putz, G.R., Scherer, R., Slater, M., Pfurtscheller, G.: Self-paced (asynchronous) BCI control of a wheelchair in virtual environments: a case study with a tetraplegic. Intell. Neuroscience 2007, 7:1–7:12 (2007)
4. Pfurtscheller, G., Müller, G.R., Pfurtscheller, J., Gerner, H.J., Rupp, R.: 'Thought' - control of functional electrical stimulation to restore hand grasp in a patient with tetraplegia. Neurosc. Lett. 351(1), 33–36 (2003)
5. Russell, J.A., Mehrabian, A.: Evidence for a three-factor theory of emotions. Journal of Research in Personality 11(3), 273–294 (1977)
6. Schmidt, L.A., Trainor, L.J.: Frontal brain electrical activity (EEG) distinguishes valence and intensity of musical emotions. Cognition and Emotion 15(4), 487–500 (2001)
7. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. Clinical Neurophysiology 113(6), 767–791 (2002)

White Blood Cell Differential Counts Using Convolutional Neural Networks for Low Resolution Images

Mehdi Habibzadeh, Adam Krzyżak, and Thomas Fevens

Department of Computer Science and Software Engineering,
Concordia University, 1455 De Maisonneuve Blvd. West,
Montréal, Québec, Canada H3G 1M8
{me_habi,krzyzak,fevens}@encs.concordia.ca

Abstract. The Complete Blood Count (CBC) is a medical diagnostic test concerned with identifying and counting basic blood cells such as red blood cells (RBC), white blood cells (WBC) and platelets. The computerized automation of CBC has been a challenging problem in medical diagnostics. In this work we describe a subcomponent system for the CBC to perform the automatic classification of WBC cells into one of five WBC types in low resolution cytological images. We describe feature extraction and consider three classifiers: a support vector machine (SVM) using standard intensity and histogram features, an SVM with features extracted by a kernel principal component analysis of the intensity and histogram features, and a convolutional neural network (CNN) which takes the entire image as input. The proposed classifiers were compared through experiments conducted on low resolution cytological images of normal blood smears. The best results were obtained with the CNN solution with recognition rates either higher or comparable to the SVM-based classifiers for all five types of WBCs.

Keywords: Complete Blood Count, white blood cell classification, convolutional neural network, support vector machine.

1 Introduction

The Complete Blood Count (CBC) [8] plays crucial role in diagnostics and management of a large number of diseases and can be ordered for a routine check-up. This test carries a large amount of vital information. CBC produces the total number of erythrocytes (Red Blood Cells) and leukocytes (White Blood Cells) in a volume of blood, the number of infected cells and indicates signs of disease. At the present time, the most reliable and complete diagnostics of blood smear is achieved in the lab by visual inspection by a medical expert of blood pathologies under optical microscope where various disorders and abnormalities in blood smears can be located and blood particles can be counted. Automatic counting systems using impedance or flow cytometry automatic counters in which blood flows through a detector [29] have also been available in medical laboratories

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 263–274, 2013.

for the past three decades. The erythrocytes (RBC) and leukocytes (WBC) that current equipment is able to classify are restricted to a few classes while requiring the use of expensive chemicals.

White blood cells are responsible for fighting infections and an WBC elevated count may indicate that patient might be suffering from inflammation, infection or stress, however a proper diagnosis requires complete knowledge about WBC present in blood smear. A normal peripheral blood sample contains five main types of WBC (the numbers in brackets indicate typical proportion of the cell type): neutrophils (40–78%), lymphocytes (25–33%), monocytes (2–8%), eosinophils (1–4%), in addition to plasma cells (0.2–2.8%) and basophil granulocytes (0–2%) [21]. Typical WBCs are shown in Fig. 1.

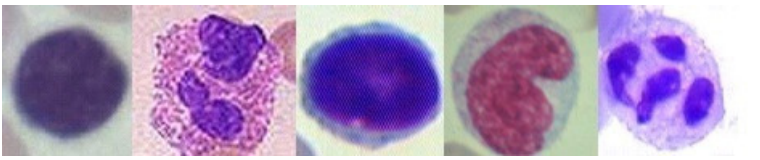


Fig. 1. WBC subtypes, left to right: Basophil, Eosinophil, Lymphocyte, Monocyte, Neutrophil

The quality of the stored blood smear images depends on the accuracy and quality of the image acquisition process. Typically an optical microscope is used in clinical setting for blood smear image acquisition. Quality of captured images is commensurate with the quality of a microscope and camera equipment, but the best high resolution instruments are costly and not available to most medical centers or labs. Typical standard quality instruments have difficulty with capturing minor details of the microscopic cell structures which can be found in WBC. Notwithstanding these image acquisition problems, a medical technician has to recognize the type specific types of white blood cells and and for this problem an automatic WBC classification system can help. The original contribution of this work aims at the development of a white blood cell differential count system using smear blood images. Such a system would perform a challenging task of efficient extraction and classification of WBC from low resolution blood smear images containing various blood cells in addition to WBCs.

2 Background and Literature Review

A primary purpose of CBC is WBC counting. The literature on automatic WBC counting in the field of image processing and pattern recognition typically deals with feature extraction, WBC segmentation and classification, e.g., [6,21,24,28]. In related work, active contours were used to track the boundaries of WBC in [20] although occluded cells were not accurately handled. Lezoray [18] proposed region-based WBC segmentation using extracted markers (or seeds) but

this approach requires prior knowledge of color information for proper seed extraction. Kumar [14] introduced a new cell edge detector while trying to accurately determine the boundary of the nucleus. A two-step segmentation framework was proposed by Sinha and Ramakrishnan [26] using k-means clustering of the data mapped to HSV color space and a neural network classifier using shape, color and texture features. In [4] WBC segmentation was achieved by means of mean-shift-based color segmentation while in [12] watershed segmentation was used.

WBC classification in the work by Hamghalam *et al.* [11] applies Otsu's thresholding for nuclei segmentation. The results are claimed to be independent of intensity differences in Giemsa-stained images of peripheral blood smears and active contours are used to extract precise boundary of the cytoplasm but in simulation and based on our recent work [8,10] the algorithm fails under different conditions. Some work, e.g., [29] segments WBCs by means of SVM classifier with a hierarchical tree-based multi-class strategy on a set of morphometrical, textural and colorimetric features. Moreover [3] introduces blood cell segmentation using conventional wavelet transformation combined with morphology in order to improve the segmentation of touching or adjacent cells. Support Vector Machine (SVM) classifier of WBC is described in [9] whereas Dual-Tree Complex Wavelet Transform is used to extract features.

In more recent work [6] in 2012 automatic differential counting is performed in two levels to segment WBC nucleus and identify the cytoplasm region. The image pre-processing with Self-Dual Multiscale Morphological Toggle (SMMT) filter is applied to gray-scale inputs to improve the accuracy and performance of two popular segmentation approaches using watershed transformation and level sets. SMMT combines scaled erosion and dilation morphological operations. Furthermore, cytoplasm region is separated by using granulometry and morphological transformations. In that work five main WBC types are classified using a K-Nearest Neighbor (K-NN) classifier with geometrical shape features and a reasonable accuracy (78% performance vs 85% classified manually by a specialist) is obtained.

3 Proposed Approach

Our work addresses the problem of classification of normal WBC cells. Image acquisition and separation of WBCs from RBCs is performed by a process from our earlier work [10]. We investigate three classification systems

1. A Support Vector Machine (SVM) [1] using intensity and histogram features (which are described in Section 3.2)
2. An SVM using intensity and histogram features reduced in dimensionality by the kernel principal component analysis (K-PCA) [13]
3. A classifier based on convolutional neural networks (CNN) [15] with automatic feature extraction

As far as we are aware, this work is the first application of CNN for the classification of cells in cytological imagery. The details of the SVM and CNN are described in Sections 3.2 and 3.3.

3.1 Image Capture and WBC Separation

Before individual WBCs are classified into their different types, the WBCs are segmented from a blood smear slide by the following procedure using the process found in [8, 10]. The process can be summarized as follows:

1. Extract sub-images containing individual closed WBC regions. The algorithm approximately determines the location of WBCs nucleus and enhances WBC boundaries
2. Use step-by-step iterative method based on RBC size estimation, circular mask, saturation value and noise removal to separate WBCs and RBCs into two individual sub-images to separate white blood cells from red blood cells.

The computational cost of the above process is primarily affected by determining an effective mask to separate the WBCs from the RBCs.

The individually separated white blood cells are saved as images in JPEG format of size 28×28 pixels to simulate the problem of WBC type recognition in images of poor quality and low magnification; see Fig. 2.

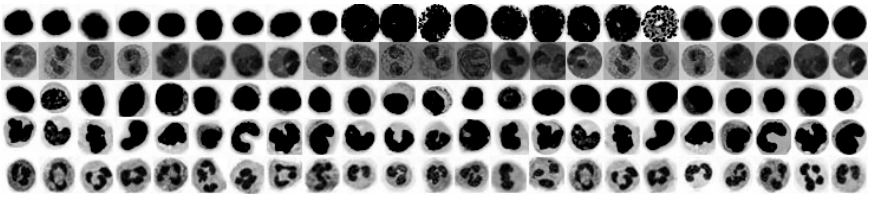


Fig. 2. WBC data, top to bottom: Basophil, Eosinophil, Lymphocyte, Monocyte, Neutrophil

3.2 SVM-Based Classifiers

Feature Extraction. After performing WBC separation using the process described above (see also [8,10]) we describe an automatic system for classification and counting of blood particles in clinical settings from a blood smear slide digital images. In the literature, three primary types of features have been used: shape, intensity and texture. Shape and geometrical features use visual differences between cells and cannot be reliably computed when the interior cell structures are not clearly visible and distinguishable in low quality images. For that reason these kinds of features are ignored in this study. Also, texture features are not considered in this paper.

Instead the following two sets of features will be investigated: intensity and histogram features, and lower dimensional features determined from the intensity and histogram features by a kernel PCA. Next we describe the features used in detail.

Intensity and Histogram Features. Consider a histogram describing the occurrence relative frequency of the intensity values of the pixels in an image. Histogram allows us to extract different properties to describe image characteristics. The histogram features that we will consider are the first four central moments of the histogram: mean, standard deviation, skewness and kurtosis. These are the first four central moments. For their precise definition, let the histogram be represented by $P(h)$ which is the relative frequency of the pixel intensity value h where $h \in [0, ..., L-1]$ with $L-1$ being the maximum intensity value. Let $I(x, y)$ be the intensity value of the pixel at row x and column y, and let N be the total number of pixels.

1. The mean gives an estimate of the average intensity level in the region of the cell:
$$\overline{h} = \sum_{h=0}^{L-1} h * P(h) = \sum_{x=0}^{Row} \sum_{y=0}^{Column} \frac{I(x,y)}{N}$$
2. The standard deviation σ is a measure of dispersion of intensity values:
$$\sigma = \sqrt{\sum_{h=0}^{L-1} (h - \overline{h})^2 * P(h)}$$
3. Skew is a measure of histogram asymmetry:
$$\frac{1}{\sigma^3} * \sum_{h=0}^{L-1} (h - \overline{h})^3 * P(h)$$
4. Kurtosis is a measure of the tail of the histogram:
$$\frac{\sum_{h=0}^{L-1} (h - \overline{h})^4}{(L-1) * \sigma^4}$$
A high kurtosis histogram has a sharper peak and longer tails, while a low kurtosis histogram has a more rounded peak and shorter thinner tail [23].

To create a larger feature vector with these histogram features, for each sample image we consider each column of pixels (y being constant above) in the image as a separate sub-image that we then compute the histogram features for. E.g., if the image is C pixels wide, we compute $4 * C$ histogram features in total for the image.

Kernel Principal Component Analysis (K-PCA) Features. Kernel PCA is used to nonlinearly project data by means of a kernel in the feature space and reduce

high dimensionality of the feature vectors. The K-PCA is an extension of the linear PCA. It is used to generate new set of features from the above intensity and histogram features, as follows. Let $(x_1, \ldots, x_n)$ is a set of features in the input space. Let the function $\phi(\cdot)$ be a non-trivial mapping to M dimensional *feature space* [22] where $M > n$. We choose a centered $\phi(\cdot)$ such that the mean of the mapped features $\phi(x_i)$ is zero. Kernel PCA performs the typical linear PCA in the feature space corresponding to the non-linear $n \times n$ kernel implementation $K = k(\cdot, \cdot)$ by the inner product between two mapped features in the feature space, $k(x, y) =< \phi(x), \phi(y) >$. For the kernel K, we determine its eigenvectors $[a_{i1}, \ldots, a_{in}]$ and corresponding eigenvalues λ_i where $i = 1 \ldots N$. For a lower dimensional embedding, we choose the eigenvectors a_{ij}, $i = 1 \ldots l$, $j = 1 \ldots n$, corresponding to the $l < m$ largest eigenvalues. Then the projection of a feature x is given by

$$y_i = \sum_{j=1}^{n} a_{ij} * k(x, x_j), \quad i = 1 \ldots l.$$

There are many options to be candidates for kernel implementation. Among them, three non-linear kernels are the most popular ones: Polynomial, Gaussian (exponential) and Sigmoidal (tanh). The Polynomial kernel is determined by $k(x, y) =< x, y >^D = (x \cdot y + C_1)^D$, and the Sigmoidal kernel is given by $k(x, y) = tanh(C_1 x \cdot y + C_2)$, where D is the polynomial degree, and C_1 and C_2 are constant values. The Gaussian kernel is given by $k(x, y) = \exp(-\frac{d^2(x,y)}{2\delta^2})$ where $d(x, y)$ is the Euclidean distance. A short review of PCA and its mathematical properties are discussed in [7, 13].

To create a new set of features from the intensity and histogram features, Kernel-PCA is applied with polynomial $D = 2$ with keeping first 50 highest eigenvalues to build a new feature vector.

SVM Classifier. We are dealing with a classification problem with poor quality samples which are degraded, have low resolution (28×28 pixels) and small size WBCs where the fine internal structures is barely visible. In this research we use a Support Vector Machine (SVM) classifier with the input features mentioned in the previous section. SVMs were introduced in 1995 by Vapnik et al. [5]. They are widely used due to their ability to deal with high-dimensional data and their efficiency in modeling diverse sources of data. An SVM constructs a maximum margin separating hyperplane in high-dimensional feature space. The nonlinear kernel mapping of the input space allows the construction of decision boundaries that are nonlinear in the input space. Typical kernel functions used in applications are radial basis functions (RBFs), sigmoidal kernels and polynomial kernels [15]. SVMs use a *Hard* or *Soft* margin (the latter allows the classifier to misclassify some points). An SVM is geared to solve two-class problems. Two strategies are used to extend conventional SVM to multi-class classification problems: one-against-all, and one-against-one [15]. More detailed mathematical treatment of SVM and its implementations can be found in [1, 15].

Kernel parameters have a direct impact on the decision boundary of the SVM [1]. Several kernels of radial basis function (RBF) and polynomial type were

experimented with for this current work, and the lowest degree polynomial, i.e., linear kernel (polynomial with $D = 1$) performed best for the current database. As in many other bio-informatics applications RBF and polynomial kernels lead to over-fitting in our high dimensional problem involving a large number of intensity and histogram features with a small input data set (28 samples for each of five WBC classes) [1].

Therefore, for this paper, we will use an SVM with a linear polynomial kernel function, soft-margin and one-against-all (1AA) strategy with 5-fold validation is used with intensity and histogram features, and the features generated by K-PCA. In order to optimize the SVM parameters, the data set is further divided into two subsets comprising 85% samples for training and the rest 15% for tuning and testing.

Given a small set of learning and tuning data consisting of five main classes for which each class includes 28 samples (total 5×28=140 samples). Each sample is described by 896-dimensional feature vector (28*28 (Intensity values) + 4*28 (Mean, Standard Deviation, Skewness, Kurtosis; computed on the 28 one-pixel wide columns of the image)).

3.3 CNN-Based Classifiers

Traditional manual-designed feature extractors are typically computationally intensive and need prior theoretical and practical knowledge of the problem at hand. They often cannot process raw images directly, while in classification scenario, automatic methods which can retrieve features directly from raw data are generally preferable. These trainable automatic systems solve classification problems without prior knowledge on the data. In medical images research, automatic feature extraction is still an open research topic and this work addresses this subject. In particular, we will use a CNN to do automatic feature extraction from the WBC cell images to be classified.

We will investigate Convolution Neural Networks [15] which are sensitive to the topology of the images being classified. A convolutional neural network (CNN) is a multilayer perceptron with a special topology containing more than one hidden layer. It allows for automatic feature extraction within its architecture and has as input the raw data. The CNN has been used for object recognition [17] and handwriting character recognition [15, 16, 25].

An CNN uses a feed-forward method for neurons feeding and back propagation for parameters training. The main advantage of the CNN approach is its ability to extract topological properties from the raw gray-scale image automatically and generate a prediction to classify high-dimensional patterns. An CNN is composed of two distinct parts. The first part consists of several layers that extract features from the input image pattern by a composition of convolutional and sub-sampling layers. Conceptually, visual features from local receptive fields [15] are extracted by an extended 2D convolution approach to gain the appropriate spatially local correlation present in the input images. Since the precise location of an extracted feature is inconsequent and dispensable, resolution reduction by

2 of the features is followed through the sub-sampling layers. The second distinct part categorizes the pattern into classes. In general, an CNN consists of three different layers: convolution layer, sub-sampling (max-pooling) layer and an ensemble of fully connected layers.

In the current study, we use an CNN with the architecture of *LeNet5* [15], see Fig. 3. In the first layers (properties extractors) convolutional filters in a 5×5 pixels window are applied over the image. It is highly recommended to add two blank pixels at each four directions to avoid missing real data at each border in convolution computations. The number of alternative three main layers depends on input database and can be varied between different input size to get better performance and confidence. In this work a *LeNet5* with eight layers is used (including first layer as input gray-scale image and also output layer). Each convolution layer (C-layers) has different feature maps, C_1 is composed of 6 units while C_3 has 16 and C_5 has 120 units. Also because of convolution windows size (5×5) and input size (28×28), the size of each convolution layer is defined as shown in Fig. 3: C_1 is 28×28, C_3 10×10, and C_5 is 1×1, a single neuron.

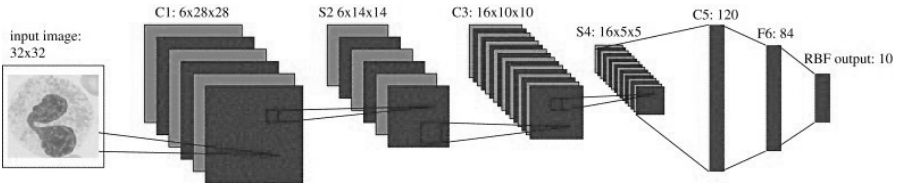


Fig. 3. LeNet-5 structure in modelling CNN for a 28×28 input image

4 Experimental Results

This section presents the WBCs classification results obtained by the proposed approaches on the existing database (115 learning samples and 25 testing ones) using three types of classifiers: SVM with intensity and histogram features, SWM with K-PCA generated features and CNN. The confusion matrices and misclassification error rates are shown in Tables 1–3.

Table 1. Confusion matrices for CNN, total over testing images

	CNN: Assigned WBC 5 classes				
Known	**Basophil**	**Eosinophil**	**Lymphocyte**	**Monocyte**	**Neutrophil**
Basophil	0.625	0.125	0.250	0.00	0.00
Eosinophil	0.00	0.95	0.05	0.00	0.00
Lymphocyte	0.125	0.00	0.875	0.00	0.00
Monocyte	0.00	0.00	0.00	0.80	0.20
Neutrophil	0.00	0.00	0.00	0.014	0.985

Table 2. Confusion matrices for Linear SVM with feature set dimensionality reduction using K-PCA, total over testing images

	Linear SVM&K-PCA: Assigned WBC 5 classes				
Known	**Basophil**	**Eosinophil**	**Lymphocyte**	**Monocyte**	**Neutrophil**
Basophil	0.60	0.00	0.30	0.10	0.00
Eosinophil	0.00	1.00	0.00	0.00	0.00
Lymphocyte	0.30	0.10	0.60	0.00	0.00
Monocyte	0.00	0.00	0.20	0.80	0.00
Neutrophil	0.10	0.00	0.20	0.00	0.70

4.1 Confusion Matrices

Since under and over counts do not clearly determine the effectiveness and validity of our frameworks, the confusion matrices comparing the relative accuracies of classifiers are introduced. The computed blood cell count results are compared with the human expert counts of the number of WBCs in each of five classes. The matrices have five rows and five columns representing the known classes of all WBC objects classified by the difference combinations of feature extraction and WBC type classification. For all available 115 (training) and 25 (testing) samples the best outcome confusion matrices for CNN (recognition rate after 105 epoch) is summarized in Table 1, linear SVM with dimension reduction using K-PCA with 2nd degree polynomial is summarized in Table 2, and linear SVM without dimensionality reduction is summarized in Table 3 below.

Table 3. Confusion matrices for Linear SVM without dimension reduction, total over testing images

	Linear SVM (without dimensionality reduction): Assigned WBC 5 classes				
Known	**Basophil**	**Eosinophil**	**Lymphocyte**	**Monocyte**	**Neutrophil**
Basophil	0.30	0.00	0.70	0.00	0.00
Eosinophil	0.00	1.00	0.00	0.00	0.00
Lymphocyte	0.40	0.10	0.50	0.00	0.00
Monocyte	0.20	0.00	0.00	0.80	0.00
Neutrophil	0.00	0.00	0.10	0.20	0.70

The tables provide confusion matrices (with normalized rows) for normal testing WBC images. In particular, for normal WBCs using CNN 85% of known WBCs were classified as such, with this classification rate decreasing to 74% for linear SVM using dimensionally reduced features using K-PCA, and to 66% for linear SVM (without K-PCA-based feature dimensionality reduction). So, based on the confusion matrices with five classes the proposed CNN classifier is much more reliable and accurate even in presence of similarity among classes (specially between *Basophil* and *Lymphocyte*) in this difficult database yielding acceptable accuracy when compared to SVM (compare the third diagonal entries in confusion matrices with *(Lymphocyte)* classification rate =87% versus 60%).

CNN yields a false positive rate (FPR) of 14%, i.e., the proportion of negatives samples incorrectly classified as positive, with this FPR increasing to 23% for linear SVM using dimensionally reduced features using K-PCA, and then to 31% for linear SVM (without using feature dimensionality reduction via K-PCA). The FPR of CNN is also smaller than the FPR of a SVM using kernel PCA and it again confirms the effectiveness of automatic feature extraction by CNN.

The CNN classifier has the best accuracy by optimizing the topological features on a difficult database containing small WBCs with no restrictions on background or capturing conditions. Another advantage of CNN it extracts features automatically while in most other classifiers the features are chosen by the designer.

5 Future Work

In this study it is assumed that the number of samples in each individual class is identical and we have a balanced database whereas in practice typical proportions of the cell types are not the same in blood smear slides (e.g., neutrophil (40- 75%) vs basophil granulocytes (0.5%)). In such cases, a Breiman Random Forest (BRF) [2] classifier may be potentially useful. The BRF algorithm can deal with imbalanced data, can handle more variables (features) than observations (large attributes, small sample), is robust for data sets containing noisy samples, and has a good predictive ability without overfitting the data. In the processing of low quality blood images we expect improved performance by combining the CNN and BRF classifiers. We are also considering other approaches such as using multidimensional pattern classifiers [27] or structural pattern recognition methods [19].

6 Conclusions

The overall objective of the current work is the development of a robust automatic system for WBC extraction and classification using low quality blood smear images. In this context, we considered a novel approach of automatic feature extraction and WBC type classification using a convolutional neural network. To test the effectiveness of CNN, a comparison was made with the more commonly used SVM classifier using standard features (with and without dimensionality reduction using K-PCA). As confusion matrices show even in case of poor samples (messy images, small and faded WBCs) WBC counts are much more accurate when the CNN classifier is used rather than an SVM classifier.

Experimental results indicate that a system based on an CNN offers an improved recognition accuracy even in presence of poor quality samples and multiple classes. It is expected that classification accuracy will be further improved by extending the data set size (especially to avoid confusion between *Basophil* & *Lymphocyte* cells since their shapes are very similar in small magnification images) and also by optimizing the CNN structure to reach higher performance in training and testing.

References

1. Ben-Hur, A., Weston, J.: A user's guide to support vector machines. In: Carugo, O., Eisenhaber, F. (eds.) Data Mining Techniques for the Life Sciences. Methods in Molecular Biology, vol. 609, pp. 223–239. Humana Press (2010)
2. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
3. Chan, H., Li-Jun, J., Jiang, B.: Wavelet transform and morphology image segmentation algorism for blood cell. In: 4th IEEE International Conference on Industrial Electronics and Applications, Xi'an, China, May 25-28, pp. 542–545 (2009)
4. Comaniciu, D., Meer, P.: Cell image segmentation for diagnostic pathology. In: Advanced Algorithmic Approaches to Medical Image Segmentation, pp. 541–558. Springer, New York (2002)
5. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
6. Dorini, L.B., Minetto, R., Leite, N.: Semi-automatic white blood cell segmentation based on multiscale analysis. IEEE Transactions on Information Technology in Biomedicine (2012) (to appear)
7. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience (November 2001)
8. Habibzadeh, M., Krzyżak, A., Fevens, T.: Application of pattern recognition techniques for the analysis of thin blood smear images. Journal of Medical Informatics & Technologies 18, 29–40 (2011)
9. Habibzadeh, M., Krzyżak, A., Fevens, T.: Analysis of white blood cell differential counts using dual-tree complex wavelet transform and support vector machine classifier. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2012. LNCS, vol. 7594, pp. 414–422. Springer, Heidelberg (2012)
10. Habibzadeh, M., Krzyżak, A., Fevens, T., Sadr, A.: Counting of RBCs and WBCs in noisy normal blood smear microscopic images. In: SPIE Medical Imaging: Computer-Aided Diagnosis, Orlando, FL, USA, February 12-17, vol. 7963, p. 79633I (2011)
11. Hamghalam, M., Motameni, M., Kelishomi, A.E.: Leukocyte segmentation in giemsa-stained image of peripheral blood smears based on active contour. In: IEEE International Conference on Signal Processing Systems, Los Alamitos, CA, USA, May 15-17, pp. 103–106 (2009)
12. Jiang, K., Liao, Q.-M., Dai, S.-Y.: A novel white blood cell segmentation scheme using scale-space filtering and watershed clustering. In: IEEE International Conference on Machine Learning and Cybernetics, Xi'an, China, November 2-5, pp. 2820–2825 (2003)
13. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer-Verlag, New York Inc. (2002)
14. Kumar, B.R., Joseph, D.K., Sreenivas, T.V.: Teager energy based blood cell segmentation. In: 14th International Conference on Digital Signal Processing, Santorini, Greece, July 1-3, pp. 619–622 (2002)
15. Lauer, F., Suen, C.Y., Bloch, G.: A trainable feature extractor for handwritten digit recognition. Pattern Recognition 40(6), 1816–1824 (2007)
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
17. LeCun, Y., Huang, F.-J., Bottou, L.: Learning methods for generic object recognition with invariance to pose and lighting. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, USA, June 27-July 2, vol. 2, pp. II-97–II-104 (2004)

18. Lezoray, O., Elmoataz, A., Cardot, H., Gougeon, G., Lecluse, M., Elie, H., Revenu, M.: Segmentation of cytological images using color and mathematical morphology. Acta Stereologica 18(1), 1–14 (1999)
19. Ogiela, M.R., Tadeusiewicz, R.: Syntactic reasoning and pattern recognition for analysis of coronary artery images. Artif. Intell. Med. 26(1-2), 145–159 (2002)
20. Ongun, G., Halici, U., Leblebicioglu, K., Atalay, V., Beksac, M., Beksac, S.: Feature extraction and classification of blood cells for an automated differential blood count system. In: International Joint Conference on Neural Networks, Washington, DC, USA, July 15-19, pp. 2461–2466 (2001)
21. Ramoser, H., Laurain, V., Bischof, H., Ecker, R.: Leukocyte segmentation and classification in blood-smear images. In: 27th IEEE Annual Conference Engineering in Medicine and Biology, Shanghai, China, September 1-4, pp. 3371–3374 (2005)
22. Rathi, Y., Dambreville, S., Tannenbaum, A.: Statistical shape analysis using kernel PCA. In: SPIE Conferences: IS&T Electronic Imaging, San Jose, CA, Jan. 15-19, vol. 6064, p. 60641B (2006)
23. Rodenacker, K., Bengtsson, E.: A feature set for cytometry on digitized microscopic images. Analytical Cellular Pathology 25(1), 1–36 (2001)
24. Shitong, W., Min, W.: A new detection algorithm (NDA) based on fuzzy cellular neural networks for white blood cell detection. IEEE Trans. on Information Technology in Biomedicine 10(1), 5–10 (2006)
25. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: 7th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 3-6, pp. 958–963 (2003)
26. Sinha, N., Ramakrishnan, A.G.: Automation of differential blood count. In: IEEE Inter'l Conf. on Convergent Technologies for Asia-Pacific Region, Bangalore, India, October 15-17, pp. 547–551 (2003)
27. Skubalska-Rafajłowicz, E.: Pattern recognition algorithms based on space-filling curves and orthogonal expansions. IEEE Transactions on Information Theory 47(5), 1915–1927 (2001)
28. Theera-Umpon, N., Dhompongsa, S.: Morphological Granulometric Features of Nucleus in Automatic Bone Marrow White Blood Cell Classification. IEEE Transactions on Information Technology in Biomedicine 11(3), 353–359 (2007)
29. Ushizima, D.M., Lorena, A.C., de Carvalho, A.C.P.L.F.: Support Vector Machines Applied to White Blood Cell Recognition. In: 5th International Conference on Hybrid Intelligent Systems, November 6-9, pp. 379–384. Rio de Janeiro, Brazil (2005)

Users Verification Based on Palm-Prints and Hand Geometry with Hidden Markov Models

Mariusz Kubanek, Dorota Smorawa, and Lukasz Adrjanowicz

Czestochowa University of Technology
Institute of Computer and Information Science
Dabrowskiego Street 73, 42-200 Czestochowa, Poland
{mariusz.kubanek,dorota.smorawa,lukasz.adrjanowicz}@icis.pcz.pl

Abstract. This paper presents a biometric system implementing identity verification on the basis of the geometry of a hand and palm-prints. Hidden Markov Models were used here as teaching and verifying tools. The paper includes suggestions of our own algorithms as well as hand features extraction methods and their specific coding for Hidden Markov Models. The obtained results show that human hand carries a lot of useful information and appropriately modified can become an interesting proposal among biometric systems, particularly carrying out the verification process with a limited number of users.

Keywords: palm-prints, hand geometry, biometrics, verification, HMM.

1 Introduction

People identification is an important problem in many computer systems and electronic systems. The well-known methods of identification, such as entering a PIN number, entering login and password or using the ID cards have many difficulties and disadvantages. It is easy to forget the PIN numbers, passwords, as well as lose identification card. In addition, the card can be stolen and protecting passwords broken. Therefore, the traditional methods of identifying people are becoming less popular. On the other hand biometric methods are gaining vast popularity in identification and verification of people. These methods use the digital measurement of certain physical and behavioral characteristics of humans and compare them with the pattern stored in the database. Until now, many biometric methods which enable identification and verification of people have been developed. From among all of these methods, hand geometry and palm-prints recognition deserves special attention.

Human hands are used as one of the crucial features in people identification. The paper [1] describes a number of features that can be extracted from a hand image and used in the identification or verification of a person. To these features the author includes, inter alia, a hand geometry which consists of measuring height, width and surface of a hand. Next, to the main features the author

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 275–285, 2013.

includes palm-prints, wrinkles, which are less regular than fingerprints, minutiae which are similar to the features obtained from fingerprints.

In another paper [2] the author shows the way of identification which is based only on basis metacarpalis. A person is identified on the basis of a hand part which is located between the wrist and the fingers. Image of the hand, as in most of the works of this type, is subjected to the segmentation which is the extraction of palm-prints. In the following work over the image, the author uses Gabor filters to obtain a more accurate frame of the palm-prints. The final step is to compare the image in the database with the received image of the hand. Studies were conducted on about 200 people. There were taken 20 pictures of left and 20 pictures of right hand of each person participating in the studies. A rejection rate of entitled person was only FRR=3%, and FAR rate=0.1%.

One of the oldest biometric techniques is a geometric measurement of the hand. This method involves typical features like the length of fingers and base of a hand, width of fingers in certain places, thickness of fingers and hands, and lines between fingers. Information about the shape of a hand are calculated on the basis of hand picture. The standard device for geometric measurement of a hand consists of a closed-circuit television camera, an infrared floodlight and a special plate which is equipped with five rings for hand positioning. The system automatically checks the correct hand position on the plate, if it is correct it takes a picture. In addition, the plate is equipped with two small mirrors in order to receive the image of a hand and thumb edges.

After taking the picture, different techniques are used for image processing and determination of the hand geometric features [3]. In some publications we can find various lists of features. For example in paper [4], the author suggests a list of 25 features with 10 teaching images for each person. The database includes 20 people. For determination of the palm features is used normal distribution - Gaussian distribution. On the other hand the authors of the paper [5] suggested a set of 16 geometrical features of hands. The set includes width and length of fingers, the ratio of size of the hand base and fingers, as well as thickness of the hand. For working over the image they used Euclidean and Mahalanobis method.

The studies of the hand image often differ from each other only in determination of the geometric features. In some papers we may find information that author suggests a set of 25 features and sometimes even 40. Obtained features undergo the process of qualification using conventional methods such as neural networks and stochastic tools. We suggest using hand shape analysis combined with palm-prints analysis to verify the identity of people. The Hidden Markov Models will be applied as a teaching-verifying tool.

2 Detection of Characteristic Hand Features

Prior to building the system that performs identity verification based on features provided by hand picture it is necessary to determine in what conditions and in what environment the system will work. For the analysis of hand shape may

be used a specially prepared plate with pins placed on the surface of it. The pins should provide the same arrangements on the plate for each of analyzed hands, which at a later stage would greatly facilitate the process of measuring geometric features of a hand. Additionally if an information on palm-prints will be required apart from the hand shape, then the arrangement of a hand on the plate will depend on a type of device receiving the image.

The obtained hand image must undergo the indispensable pretreatment processes in order to adjust the image to the requirements of the measurement procedure of geometric features. First, detection of a hand in the image is performed. It is obvious that the hand picture was taken on a specially prepared plate of uniform surface colour, different from the hand colour. Thresholding can be done by cutting out the surface colour from the image and replacing it with e.g. black colour (for monochrome pictures), or by a specific skin colour (for colour pictures).

The determined area of the hand easily enables further analysis of its geometric features, including accurate measurement of each analyzed hand. Because special pins are placed on the plate, it is sufficient to take the distance between any pins as a reference point of dimension, which allows to automatically scale the image to identical actual sizes, regardless of the distance from the camera. The proceedings pattern during the measurement of the hand geometry and setting the limited hand area are shown in Fig. 1.

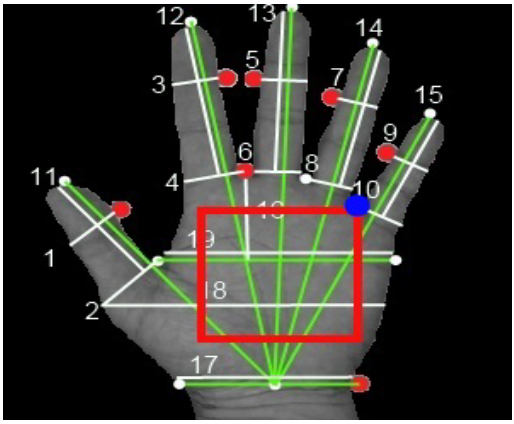


Fig. 1. The method of geometric measurement of hand features

On the basis of pins placed on the plate (marked in red in the picture) it is possible to detect additional points needed for assumed measurements (white dots). It is performed in the following way, for example, a thumb is searched for the maximum non-zero pixel (not black) to the left of the pin and the minimum non-zero pixel to the right of the pin. For the other fingers, the method is very similar. The points characterizing beginnings and ends of fingers are marked in this way. The last thing to do is to designate points which characterize width of

a hand and wrist and half the width of the wrist. For this purpose the auxiliary lines are drawn (green) parallel to the edge of the photo from the pin placed at the wrist and from the minimum point on the right side of the thumb. The intersection of the auxiliary lines and the hand edge designates the remaining points. Auxiliary lines are also drawn from the fingertips to the point which indicates half the width of the wrist (also green). The prepared frame of auxiliary lines will allow to make further measurements.

Determination of the sizes is based on the assumption that the measurements are carried out in directions perpendicular to the auxiliary lines at the designated characteristic points. In this way determined are: the width of a thumb (distance 1 and 2) and the length of a thumb (distance 11), the width of an index finger (distance 3 and 4) and the length of an index finger (distance 12), the width of a middle finger (distance 5 and 6) and the length of a middle finger (distance 13), the width of a ring finger (distance 7 and 8) and the length of a ring finger (distance 14), the width of a small finger (distance 9 and 10) and the length of a small finger (distance 15), the distance between the base pin and line 19 characterizing the width of a hand (distance 16), the width of a wrist (distance 17), the width of a hand (distance 18 and 19). Nineteen different distances characterizing geometric features of analyzed hand are designated in this way. Due to the fact that the distance between the leading pins located between index finger and middle finger is exactly 10 millimeters, all determined distances can also be reported in millimeters.

Since there is a possibility of measuring geometric features of a hand in a millimeter scale, it is possible each time to mark out e.g. a square or rectangular area of pre-defined dimensions. The reference point may be the end of a ring finger and a small finger (see Fig. 1).

Searching for simplicity in proceedings, and thus accepting only the main palm-prints analysis, an interesting solution could be the use of a simple directional edge detector. With the appropriate settings of detector coefficients it is possible to omit little, short, faint lines. However, there is a problem because such a clearly visible hand line will have designated edges on both its sides, which does not facilitate further work. However, at this time we can use a morphological operation - dilatation, which will connect the separate edge lines.

Dilatation greatly increases the thickness of lines, but the shape and location of major palm-prints is not changed. In order to interchangeably define the palm-prints shapes, these bold elements should be brought closer by one pixel thick lines. This can be done by using the linear Hough transform. The transform allows the detection of the straight lines in binary image. With appropriate parameters (the minimum length of searched palm-prints should be adjusted) the output effect of Hough transform will be the whole family of lines gathered around the bold palm-prints. Taking into account the distribution of nodes in the Hough space, we select only those lines which nodes are arranged in a continuous and intense way. Palm-prints which are selected in this way are also defined with an approximating line. Fig. 2 illustrates the effect of designating major palm-prints with the usage of the linear Hough transform.

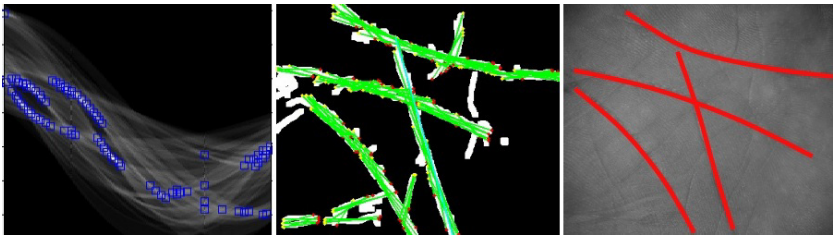


Fig. 2. Designated space of Hough transform on the left, applied operation of lines detection on the basis of the transform in the middle and approximation operation on the right

3 Coding of Appointed Features for Hidden Markov Models

Coding of hand features is connected with determining the way of procedure with each hand feature. First from appointed features concerns the geometric shapes of the hand. Forming of the observation vector is combined with giving consecutive geometric dimensions which are calculated for each finger and the whole hand. Calculated 19 dimensions in millimetres are contained in the range of the whole numbers from a to b. The observation vector will be composed of other dimensions. It will allow to distinguish the two hands of similar shapes but different in sizes.

Coding of palm-prints of the hand can be executed starting from dividing the input image into a fixed number of square sub-images. At this work the size of image consisting of palm-prints is 60 x 60 mm. At these values the input image can be divided into 144 sub-images.

Each sub-image has one observation symbol. Coding of palm-prints consists in the choice of only these sub-images which consist any of fixed palm-prints. Forming the observation vector the sub-images are chosen according to approved numeration. If a given sub-image consists a line, then besides the number of sub-image, in the observation vector the information about the average angle where a given piece of a line is arranged relative to the horizontal axis. In order to limit the observation symbols connected with coding the angles, defined between palm-prints and the horizontal axis, the division of semi-circle (180^o) is accepted to a fixed number of slices (e.g. 16). In such a way for every angle one of 16 observation values are assigned. The scheme of such coding is shown in Fig. 3. Based on the above mentioned scheme, the sample observation vector begins from the third value for which the angle code is 13, the next observation is 16-th value with the angle code 13, and in conse-quence the next is 17-th with the angle code 14, etc. If in the given sub-image there are two or more lines, then besides the number of an observed sub-area, the angle codes of all lines are assigned (starting from the lowest lines). The resultant angle is assigned on the basis of a tangent to a curve (describing the palm print) in points P1 (the beginning of the line), P2

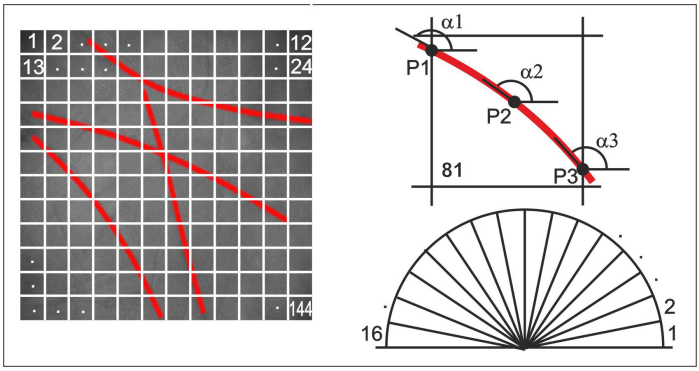


Fig. 3. The scheme of procedure during assigning the observation vectors using this coding method

(the middle of the line) and P3 (the end of the line) within a given sub-image. In the next steps the average angle is determined from the angles $\alpha 1$, $\alpha 2$ and $\alpha 3$.

The presented way of coding assumes the possibility of erroneous detection of main palm-prints, which determines an incomplete line or its entire omission. The essential element in such a way of coding is the acceptable variable number of observation which in case of vector measurement using the distance method, would introduce the complement of vector size (e.g. unities, zeros), causing too much similitude of data. Using the teaching-verifying tool in the case of Hidden Markov Models, the possibility of generalizing the patterns is kept, despite rapidly changing observation vectors of certain characteristics of the image.

4 The Use of Hidden Markov Models

The Hidden Markov Model (HMM) is a powerful statistical tool for modeling generative sequences that can be characterised by an underlying process generating an observable sequence. HMMs have found application in many areas interested in signal processing, and in particular speech processing, but have also been applied with success to low level natural language processing tasks such as part of speech tagging, phrase chunking, and extracting target information from documents. HMMs are stochastic models and are widely used for characterizing the spectral properties of frames of patterns [6–11].

A HMM is characterized by: the number of states in the model N, the number of Gaussian mixtures per state M, the state transition probability distribution A, the observation symbol probability distribution B, and the initial state distribution π. The compact notation $\lambda = (A, B, \pi)$ is used to indicate the complete parameter set of an HMM model.

HMMs have three fundamental problems, namely recognition, segmentation and trying. These problems can be defined as follows: 1) recognition problem

is computing the probability $P(O|\lambda)$ given the observation sequence O and the model λ, 2) segmentation problem is the determination of the optimal state sequence given the observation sequence $O = O1, O2, , OT$, and the model λ, 3) training problem is the adjustment of model parameters $\lambda = (A, B, \pi)$ so as to best account for the model states, this is equal to adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$.

Given a HMM, and a sequence of observations, we'd like to be able to compute $P(O|\lambda)$, the probability of the observation sequence given a model. This problem could be viewed as one of evaluating how well a model predicts a given observation sequence, and thus allow us to choose the most appropriate model from a set. The probability of the observations O for a specific state sequence Q is:

$$P(O|Q,\lambda) = \prod_{t=1}^{T} P(o_t|q_t,\lambda) = b_{q1}(o_1) \times b_{q2}(o_2)...b_{qT}(o_T) \tag{1}$$

and the probability of the state sequence is:

$$P(Q|\lambda) = \pi_{q1} a_{q1q2} a_{q2q2}...a_{qT-1qT} \tag{2}$$

so we can calculate the probability of the observations given the model as:

$$P(Q|\lambda) = \sum_{Q} P(O|Q,\lambda)P(Q|\lambda) = \sum_{q_1...q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1q_2} b_{q_2}(o_2)...a_{q_T-1q_T} b_{q_T}(o_T) \tag{3}$$

This result allows the evaluation of the probability of O, but to evaluate it directly would be exponential in T.

A better approach is to recognise that many redundant calculations would be made by directly evaluating eq. 3, and therefore caching calculations can lead to reduced complexity. We implement the cache as a trellis of states at each time step, calculating the cached valued (called α) for each state as a sum over all states at the previous time step. α is the probability of the partial observation sequence $o_1, o_2, ..., o_t$ and state s_i at time t. We define the forward probability variable:

$$\alpha_t(i) = P(o_1 o_2 ... o_t, q_t = s_i \lambda) \tag{4}$$

so if we work through the trellis filling in the values of α the sum of the final column of the trellis will equal the probability of the observation sequence. The algorithm for this process is called the forward algorithm and is as follows: Initialisation:

$$\alpha_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N \tag{5}$$

Induction:

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] b_j(o_{t+1}), 1 \leq t \leq T-1, 1 \leq j \leq N \tag{6}$$

Termination:

$$P(Q|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{7}$$

For each state s_j, $\alpha_j(t)$ stores the probability of arriving in that state having observed the observation sequence up until time t.

It is apparent that by caching α values the forward algorithm reduces the complexity of calculations involved to N^2T rather than $2TN^T$. We can also define an analogous backwards algorithm which is the exact reverse of the forwards algorithm with the backwards variable:

$$\beta_t(i) = P(o_{t+1}o_{t+2}...o_T|q_t = s_i, \lambda) \tag{8}$$

as the probability of the partial observation sequence from $t + 1$ to T, starting in state s_i [7].

5 Experimental Results

The research to determine the effectiveness of the identity verification on the basis of the hand shape and main palm-prints are conducted for the usefulness of Hidden Markov Models also to such biometric systems. The hand geometry provides 19 different dimensions describing the shape of a given user. If the identity verification is on the basis of the hand shape, then the comparative method is enough to estimate the distance between two compared vector features.

The first of the research connected with the human hands is to determine the correctness of the identity verification on the basis of hand geometry. The study was to determine the false rejection rate and the false acceptance rate for different number of analysed users. 500 different users were tested for whom 2 photos were downloaded (first for the registration and second for testing). The error rates were determined for 100, 200, 300, 400 and 500 users. All pictures which form input data were taken in the same conditions and using the same equipment. The results of errors are in the Table. 1. The further tests concerned the influence of the use of Hidden Markov Models on the quality of identity verification. The way of connecting hand features provided by the shape and by the main palm-prints are considered. It is not easy to choose the ideal method based on the characteristics which while having a lot of users show quite a similarity. The twofold way of verification was presented. First one was based on the use of Hidden Markov Models only to analysis of the palm-prints, the shape features were left to vector comparison. The second way accepted both the geometric characteristics and palm-prints characteristics as the input data during forming of the observation vector. In the first case the correct verification required the proper answer from the camera which compares the distance between vectors and the camera which carries out the recognition. Only mutual compatibility designs the correct verification. The results of this verification are in the Table. 2, where the error rates are determined using the number of users of 100, 200, 300, 400 and 500. Better solution is to allow for the geometric hand features during forming of the observation vectors given on the input models during teaching and verifying. The more distinguishable features are provided to the model the better results are. Of course the correctness of assigning and coding of input data is important. In the Table. 3 there are the results of assigning the errors during user

Table 1. Results of tests for identity verification based on the geometry of the hand

Tests	Num. of Users	Num. of FA	FAR [%]	Num. of FR	FRR [%]
1	100	0	0.00	0	0.00
2	200	1	0.50	4	2.00
3	300	6	2.00	21	7.00
4	400	13	3.25	39	9.75
5	500	26	5.20	73	14.60

Table 2. Results of tests for identity verification based on the geometry of the hand and palmprint with separate HMM

Tests	Num. of Users	Num. of FA	FAR [%]	Num. of FR	FRR [%]
1	100	0	0.00	2	2.00
2	200	0	0.00	7	3.50
3	300	3	1.00	13	4.33
4	400	9	2.25	27	6.75
5	500	16	3.20	55	11.00

Table 3. Results of tests for identity verification based on the geometry of the hand and palmprint with coupled HMM

Tests	Num. of Users	Num. of FA	FAR [%]	Num. of FR	FRR [%]
1	100	0	0.00	0	0.00
2	200	0	0.00	0	0.00
3	300	1	0.33	3	1.00
4	400	4	1.00	7	1.75
5	500	7	1.40	13	2.60

verification, based on combined vector of shape and palm-prints characteristics. The obtained results show the purpose of the use of Hidden Markov Models to increase the effectiveness of biometric system using the verification based on the hand shape and main palm-prints. If in the verification process the use of only geometric features, describing the shape of a hand, with the increase of the user numbers in the database, both the false rejection rate and the false acceptance rate increase to dangerous levels. As far as the false rejection rate for 500 users at 14.6% is only inconvenient for users, the false acceptance rate at over 5% is a serious danger for the system safety.

The use of Hidden Markov Models must have its quality justification. If you assume as input data only the characteristics connected with the image of palm-prints, and geometric features will be compared as usual vector comparison, then the false rejection rate increases in relation to increasing number of users.

If only while forming of vector observation geometric features of hand are taken into consideration then the obtained results confirm the wisdom of the use of Hidden Markov Models to such biometric systems. For 500 users the false acceptance rate at 1.4% is not enough. Also the false rejection rate at 2.6% is

not a great hindrance to potential users of the system. For less than 100 users the system is very safe and can be used to secure important resources.

The perfect solution would be to gain the index at 0.0001% however it was not the main purpose of the analysis in this chapter. Taking into consideration the geometric features of a hand seen from above and geometric features of a hand seen from the side and besides main palm-prints, also detailed palm-prints then the system could be formed which would function as the identification system. Most verifying systems based on biometric features are dedicated for the small number of users, so the fulfilling of level of security is not a complicated task. The searching of such solutions is important which at relatively low effort and using useful tools increase the effectiveness of biometric systems [12, 13]. The table of the obtained results of above studies is shown in Fig. 4.

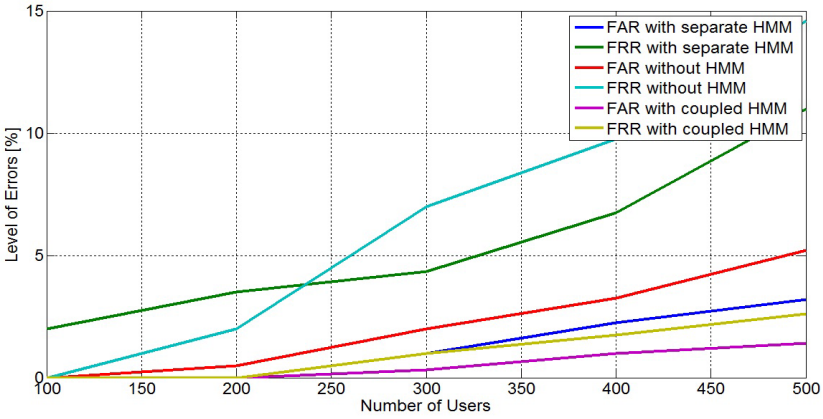


Fig. 4. The graph of dependence of false acceptance rate and false rejection rate on the number of users in the database (without, with separate and combined HMM)

6 Conclusion and Future Work

In this article the biometric system for identity verification is shown on the basis of hand shape and main palm-prints. As the tool which supports the process of verification Hidden Markov Models are presented. Several solutions are proposed connected with the analysis of the image for extraction of searching features and the way of connecting assigned geometric and visual features. The author‘s way of coding of palm-prints was described, developed for the need and possibility of the use of Hidden Markov Models.

Conducted studies show that in case of big number of users the analysis of geometric features gives not enough safety. But if you take into consideration the features given by the picture of the palm then by proper coding it is possible to gain better results.

Further studies will be connected with increasing the effectiveness of biometric system through the expansion of the number of variable characteristics to geometric features provided by the side image of the hand and visual features coming from the detailed palm-prints.

References

1. Shu, W., Zhang, D.: Automated Personal Identification by Palmprint. Optical Engineering 37, 2359–2362 (1998)
2. Fatric, I., Ribaric, S.: Colour-Based Palmprint Verification – An Experiment. In: Watteyne, T., Yazidi, A. (eds.) The 14th IEEE Mediterranean Electrotechnical Conference Proceedings, pp. 890–895. IEEE, Ajaccio (2008)
3. Guo, J., Liu, Y., Yuan, W.: Palmprint Recognition Using Block Entropy Map From a Single Image Per Person. Journal of Computational Information Systems 8(12), 5129–5135 (2012)
4. Sanches-Reillo, R., Sanchez-Avila, C., Gonzalez-Marcos, A.: Biometric Identification through Hand Geometry Measurements. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(10), 1168–1171 (2000)
5. Jain, A.K., Ross, A., Pankanti, S.: A Prototype Hand Geometry Based Verification System. In: Proc. of 2nd Int. Conference on Audio and Video-Based Biometric Person Authentication, Washington D.C, pp. 166–171 (1999)
6. Rabiner, L., Yuang, B.H.: Fundamentals of Speech Recognition. Prentice Hall Signal Processing Series (1993)
7. Kaynak, M.N.N., Zhi, Q., Cheok, A.D., Sengupta, K., Chung, K.C.: Audio-Visual Modeling for Bimodal Speech Recognition. In: Proc. 2001 International FuzzySystems Conference (2001)
8. Blunsom, P.: Hidden Markov Models, `http://digital.cs.usu.edu/~cyan/CS7960/hmm-tutorial.pdf`
9. Bobulski, J.: Selection of Parameters of HMM. Measurement, Automation, Control 10, 844–846 (2009)
10. Bobulski, J.: Analysis of 2D Problem in HMM. Methods of Applied Informatics. 4(17), 27–33 (2008)
11. Verma, N.K., Hanmandlu, M.: Additive and Non-additive Fuzzy Hidden Markov Models. IEEE Trans. Fuzzy Systems 18(1), 40–56 (2010)
12. Choras, M., Kozik, R.: Contactless Palmprint and Knuckle Biometrics for Mobile Devices. Pattern Analysis and Applications 15(1), 73–85 (2012)
13. Choraś, M., Kozik, R.: Fast Feature Extractors for Palmprint Biometrics. In: Chaki, N., Cortesi, A., et al. (eds.) CISIM 2011. CCIS, vol. 245, pp. 121–127. Springer, Heidelberg (2011)

A New Approach to Determine Three-Dimensional Facial Landmarks

Sebastian Pabiasz and Janusz T. Starczewski

Institute of Computational Intelligence,
Czestochowa University of Technology, Czestochowa, Poland
{sebastian.pabiasz,janusz.starczewski}@iisi.pcz.pl

Abstract. In recent years, the number of biometric solutions based on 3D face images has increased rapidly. Such solutions provide a much more accurate alternative to those using flat images; however, they are much more complex. The underlined assumption is that a human can distinguish unique features of a face. To date, researchers have not determined which features are the best to be chosen for automatic face recognition. This article presents the preliminary results of research on an alternative form of representation of the face, the application in biometric systems. The representation, is an extension to anthropometric landmarks.

Keywords: biometric, 3D face, mesh, depth map.

1 Introduction

A biometric system is a pattern recognition system that determines the authenticity of an individual using physical or behavioral features. The physical features include unique anatomical features such as fingerprint, DNA, etc. Behavioral features are related to the behavior of a person e.g. signature[4][5]. Biometric systems are divided into two groups. First, the systems that require user interaction e.g. systems based on fingerprints. A biometric capture device must scan a fingerprint, hence user intervention is required. The second group consists of systems based on the feature that is always available and easily available such as face. Then, a biometric system can work in two states: a normal operating state (identification/verification of persons), and a learning state, which is required for any pattern recognition system [12]. The social demand for security and fraud control applications, in which it is necessary to establish a personal authentication, has caused the need for increased research in such systems [13].

Face recognition is a natural human trait. We can recognize a face in complex environments; however, just biometric features are in the field of research for decades. To the present day, researchers have not answered the question which feature of the face is unique and none developed solution is not faultless. Since the 90's, human face recognition has attracted a great deal of attention in pattern recognition and computer vision. This trend can be explained by a steadily

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 286–296, 2013.

increasing demand for more secure systems and a need to identify people in an automatic way. The face is suitable for most of these solutions; however, it is a biometric feature, with the highest complexity.

Face recognition is a complex process. It consists of several stages. In the first stage, the face image is taken. Then, it is separated from the background. All of these stages are included in the pre-processing. From this moment begins a process of analysis of the face image and features selection. There are two common approaches:

- **Geometric Feature-Based Approaches.** In this group of approaches, all visible features could be analyzed. Researchers have been studying features-based approaches for decades. Example of features include the tip of the nose, the notch between the nose and the upper lips. [2] [7]
- **Holistic Approaches.** In this group of approaches, the representation of a face should have sufficient information to differentiate each pair of dissimilar face images in a compact face space. Holistic approaches are likely to be more accurate since they use significantly more information about the face. However, they are also likely to slow due to the complexity of the matching process. [9] [25] [1]

There are also mixed systems,that combine geometrical and holistic features (e.g. active appearance model, local feature analysis, elastic graph matching). One of the most common geometric features are landmarks. These are the points that always occur on the face. All are visible only in the frontal view, but there are symmetrically. Their idea is derived from anthropometry. Facial recognition systems are based on information about their location, proportion of distances among landmarks. [11] [10] [26]

After features selection, is beginning a process of verification or identification. Biometric system may also be in the stage of training. In this stage learns to recognize new users.

This paper presents a new method of determining the characteristics of the face. The method is based on a 3D model. The authors reach the basics and try to develop an alternative method in relations to generally accepted methods.

2 3D Face Model

Biometric systems may obtain information about the face in many ways. The most accurate data comes from dedicated devices such as 3D scanners. You can also obtain information from images with different lighting. In the 80's, the first works on the 3D object recognition were based on data from low-resolution detectors. Data were organized in a matrix in which each cell contained the value of the distance to the detector. This type of data is called a depth map. The data were small in size and easy to process, but not accurate. With an increasing resolution of sensors, the data began to be more accurate but processing was

more complex. Many areas in a data set contained redundant data. Therefore, researchers have started to build 3D face models. Models consist of vertices and walls, which makes it easy to transform. However, their direct comparison is a very complex operation. Three-dimensional model of the face of the same person can not be the same. The face is the object that is variable due to facial expressions.

3 Classical Landmarks

Landmarks are derived from biological knowledge, specifically from anthropometry, the science of measuring the human body. Formally, "A landmark is a point of correspondence on each object that matches between and within populations" [8]. A human face is usually made up of 59 landmarks. These are points that can be identified on each face. Landmarks reduce the set of points that describe the face. Research shows that there is not much difference between the use of 11 and 59 landmarks. Most existing methods for landmark detection are dependent on prior knowledge of feature map threshold, orientation and pose. Typically, the first step in the search for landmarks is segmentation. Deformable models such as Active Shape Models [3] and Active Appearance Models [14] are used extensively for image segmentation and landmark detection. The shape model used in these approaches, aims to perform image interpretation using prior statistical knowledge of the shape to be found. 3DMM is a concept closely related to AAM where a 3D model is used to estimate the 3D parameters in a 2D image and to segment the object. Some algorithms attempt to extract features of the face based on the face shape and to classify these regions based upon these results. Such approaches use differential geometry. [15] [19]

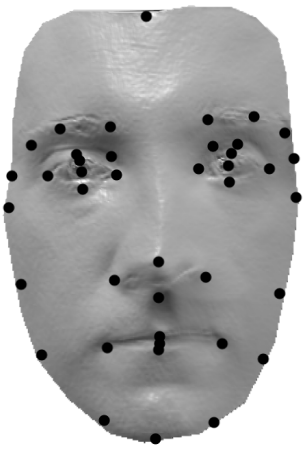


Fig. 1. Face with selected classical landmarks

3.1 Segmentation

Segmentation is the process that provides a necessary organization of data points by partitioning them into connected regions or parts that can be approximated by standard surfaces (e.g. planes, cylinders) or volumetric primitives (e.g. super ellipsoids).

	Hyperbolic	Convex cylindrical	Convex elliptical	Planar	Concave cylindrical	Concave elliptical
Shape						
Sign(*H*)	+/0/-	-	-	0	+	+
Sign(*K*)	-	0	+		0	+

Fig. 2. Default shapes in HK segmentation (based on [15])

$$K = K_1 K_2, \tag{1}$$

$$H = \frac{1}{2}(K_1 + K_2), \tag{2}$$

where

$$K_1, K_2 = H \pm \sqrt{H^2 - K}(K_1 \geq K_2) \tag{3}$$

One of the most popular segmentation is performed by HK algorithm. Gaussian (K) and Mean (H) curvatures are the most widely used indicators for surface shape classification in range image analysis. Gaussian and Mean curvatures, which are calculated from the two principal curvatures *K1* and *K2*.

4 Determining Three-Dimensional Facial Landmarks

In the first stage, the input set is organized in the form of a depth-map. We have to examined the possibility to extract facial landmarks (new, no relation to anthropometric points) on the basis of extremes. We assume that each row and each column is represented in a function form. Each function can be classified as a four types of values:

- **local minimum** is a local minimum of a function at a specified processing range
- **local maximum** is a local maximum of a function at a specified processing range

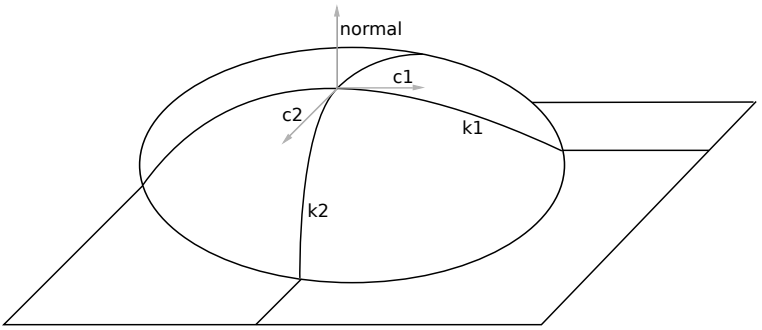


Fig. 3. Principal Curvatures and Principal Directions (based on[24])

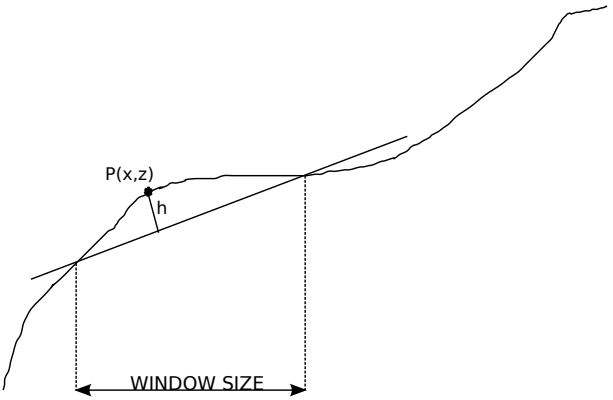


Fig. 4. Determination of the height of the point

- **global minimum** is a global minimum of a function
- **global maximum** is a global maximum of a function

Our method has two stages (Algorithm 1). First stage extracts characteristic points from columns, second from rows. In each step, only points of the selected range are analyzed.

In our algorithm, height of each point is a distance from the straight line passing through the window borders(fig. 4).

5 Experimental Materials and Results

5.1 3D Face Database

The study was based on a set of biometric three-dimensional images *NDOff-2007*[6]. The collections 6940 3D images (and corresponding 2D images) of 387 human subject faces. The advantage of this collection is that, for each person, there are several variants of face orientation.

Algorithm 1. First state of landmark extraction

```
for x = 1 → COLUMNS do
  for y = 1 → WINDOWS_SIZE do
    find_Local_Minimum
    find_Local_Maximum
    if is_Global_Minimum_in_Range then
      save_Global_Minimum
    end if
    if is_Global_Maximum_in_Range then
      save_Global_Maximum
    end if
  end for
end for
for x = 1 → ROWS do
  for y = 1 → WINDOW_SIZE do
    find_Local_Minimum
    find_Local_Maximum
    if is_Global_Minimum_in_Range then
      save_Global_Minimum
    end if
    if is_Global_Maximum_in_Range then
      save_Global_Maximum
    end if
  end for
end for
```

5.2 Results

Presented method allowed to obtain all the relevant points of the face. Figure 5, shows data collected from rows and columns. From the analysis of the lines, more points are produced but they are less suitable (tab. 1). Figure 7 shows that data collected from both parts of the algorithm result in small set of potential features which can be proceeded by the face recognition process. As it can be seen, especially highly visible points are global maxima regardless of point of view (fig. 8, fig. 6).

Fig. 5. Collected data: (a) — from rows, (b) — from columns

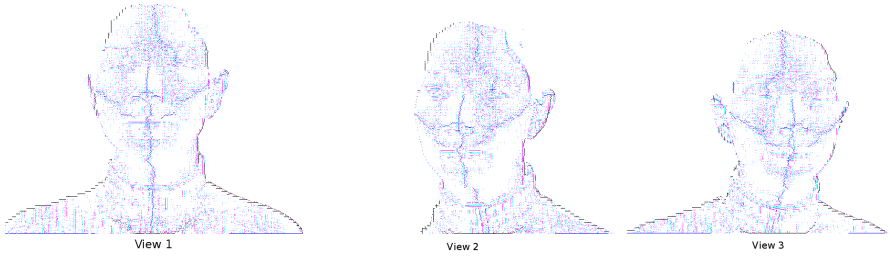


Fig. 6. Data collected from different views

Table 1. Statistics of collected points(fig. 6)

	View 1	View 2	View 3
Points	21551	17023	16550
Points from rows:	7024	5482	5088
Points from columns:	1246	814	1080

Table 2. Statistics of collected points at different range

Window size	Points	Points frow rows	Points from columns
5	21551	1246	7120
10	21551	1246	5148
20	21551	1232	3556
50	21551	1228	2336
100	21551	752	1336

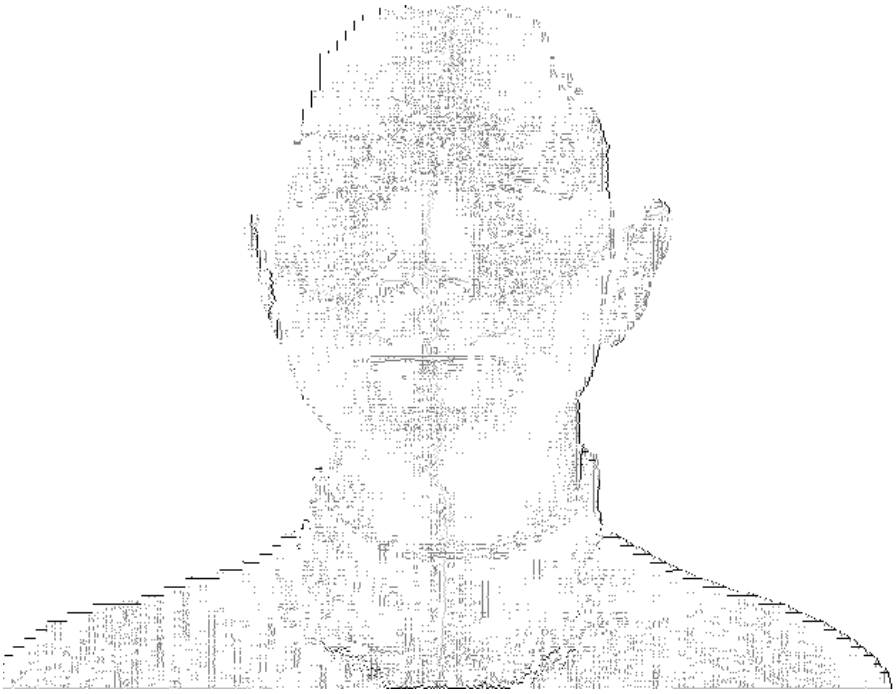


Fig. 7. Final result

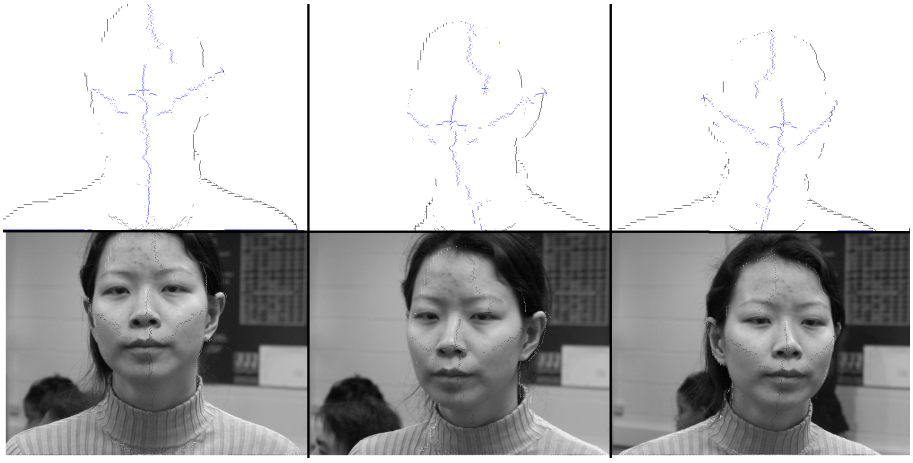


Fig. 8. Global minima and maxina from different points of view

6 Conclusion

In this contribution the preliminary results of work on the new representation of 3D face was presented. Our representation based on characteristic points. The new criterion for determining the height of point has been applied. This resulted in a better selection of points. The most interesting fact is potential use of global maxima to determine orientation of a face.

In the future work we want to focus on the further development of the representation of the face, in particular, on the methods of model interpretation as well as combinations with methods of image understanding[18][22][23]. The development of face recognition techniques, requires constant testing, so in the future studies we want to focus on face recognition using uncertain fuzzy classification techniques eg.[16][17][21][20].

References

1. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. 19(7), 711–720 (1997)
2. Brunelli, R., Poggio, T.: Face recognition: features versus templates. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(10), 1042–1052 (1993)
3. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models their training and application. Comput. Vis. Image Underst. 61(1), 38–59 (1995)
4. Zalasinski, M., Cpalka, K.: A new method of on-line signature verification using a flexible fuzzy one-class classifier. In: Selected Topics in Computer Science Applications, pp. 38–53 (2011)
5. Zalasiński, M., Cpałka, K.: Novel algorithm for the on-line signature verification. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS (LNAI), vol. 7268, pp. 362–367. Springer, Heidelberg (2012)

6. Faltemier, T.C., Bowyer, K.W., Flynn, P.J.: Rotated profile signatures for robust 3d feature detection. In: 8th IEEE International Conference on Automatic Face Gesture Recognition, FG 2008, pp. 1–7 (September 2008)
7. Huang, C.-L., Chen, C.-W.: Human facial feature extraction for face interpretation and recognition. In: Proceedings of 11th IAPR International Conference on Pattern Recognition, Conference B: Pattern Recognition Methodology and Systems, August 3-September, vol. II, pp. 204–207 (1992)
8. Mardia, K.V., Dreden, I.L.: Statistical Shape Analysis. John Wiley, New York (1998)
9. Kirby, M., Sirovich, L.: Application of the karhunen-loeve procedure for the characterization of human faces. IEEE Trans. Pattern Anal. Mach. Intell. 12(1), 103–108 (1990)
10. Kong, S.G., Heo, J., Abidi, B.R., Paik, J., Abidi, M.A.: Recent advances in visual and infrared face recognition - a review. Computer Vision and Image Understanding 97, 103–135 (2005)
11. Lanitis, A., Taylor, C.J., Cootes, T.F.: An automatic face identification system using flexible appearance models. Image and Vision Computing 13, 393–401 (1995)
12. Li, S.Z.: Encyclopedia of Biometrics, 1st edn. Springer Publishing Company, Incorporated (2009)
13. Martínez, A.M., Yang, M.-H., Kriegman, D.J.: Special issue on face recognition. Comput. Vis. Image Underst. 91(1-2), 1–5 (2003)
14. Matthews, I., Baker, S.: Active appearance models revisited. Int. J. Comput. Vision 60(2), 135–164 (2004)
15. Moreno, A.B., Snchez, N., Fras-Martnez, E.: Robust representation of 3d faces for recognition. IJPRAI 20(8), 1159–1186 (2006)
16. Nowicki, R.: On classification with missing data using rough-neuro-fuzzy systems. Applied Mathematics and Computer Science 20(1), 55–67 (2010)
17. Nowicki, R., Rutkowski, L.: Soft techniques for bayesian classification. In: Rutkowski, Kacprzyk (eds.) Proceedings of the Sixth International Conference on Neural Network and Soft Computing, Zakopane, Poland, June 11-15, Advances in Soft Computing, pp. 537–544. Springer Physica-Verlag (2003)
18. Ogiela, M.R., Tadeusiewicz, R.: Artificial intelligence structural imaging techniques in visual pattern analysis and medical data understanding. Pattern Recognition 36(10), 2441–2452 (2003)
19. Rohr, K.: On 3d differential operators for detecting point landmarks. Image and Vision Computing 15(3), 219–233 (1997)
20. Scherer, R., Rutkowski, L.: A fuzzy relational system with linguistic antecedent certainty factors. In: Rutkowski, Kacprzyk (eds.) Proceedings of the Sixth International Conference on Neural Network and Soft Computing, Zakopane, Poland, June 11-15. Advances in Soft Computing, pp. 563–569. Springer Physica-Verlag (2003)
21. Scherer, R., Rutkowski, L.: Neuro-fuzzy relational classifiers. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 376–380. Springer, Heidelberg (2004)
22. Tadeusiewicz, R., Ogiela, L., Ogiela, M.R.: The automatic understanding approach to systems analysis and design. International Journal of Information Management 28(1), 38–48 (2008)

23. Tadeusiewicz, R., Ogiela, M.: Why automatic understanding? In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4432, pp. 477–491. Springer, Heidelberg (2007)
24. Tanaka, H.T., Ikeda, M., Chiaki, H.: Curvature-based face surface recognition using spherical correlation. principal directions for curved object recognition. In: Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 372–377 (April 1998)
25. Turk, M.A., Pentland, A.P.: Face recognition using eigenfaces. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 1991, pp. 586–591 (June 1991)
26. Wiskott, L., Fellous, J.-M., Kuiger, N., von der Malsburg, C.: Face recognition by elastic bunch graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 775–779 (1997)

A Comparison of Dimensionality Reduction Techniques in Virtual Screening

Kitsuchart Pasupa

Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
kitsuchart@it.kmitl.ac.th

Abstract. Most of the screening methods have always struggled to deal with the high dimensionality of data in virtual screening task. One of the most commonly used techniques to reduce the high dimensional data is principal component analysis (PCA). PCA and its variants have been introduced and re-introduced to solve the problems in particular tasks in real world applications. In this paper, PCA and four variants of it are compared and analyzed together in virtual screening task in particular using fingerprint representation. Fingerprint is one of the most regularly used descriptors in virtual screening task. None of these methods have never been compared and studied together with high dimensional and binary-valued data elsewhere. The results show superiority of the variants of PCA to PCA on the most heterogeneous classes, while the methods are competitive to PCA on the homogeneous classes. Supervised PCA is found to be the best technique and is competitive to Fisher discriminant analysis. It should be noted that Fisher discriminant analysis uses all the provided information while Supervised PCA uses only few components.

Keywords: dimensionality reduction, principal component analysis, virtual screening.

1 Introduction

Virtual Screening (VS) is a set of computational methods or *in silico* analogues of biological screening. The aim of VS is to score, rank and/or filter a set of chemical structures using one or more computational procedures in order to ensure those molecules with the largest prior probabilities of activity are assayed first in a "lead discovery program"[1]. The analysis and manipulation of chemical structure can be carried out through the use of molecular descriptors. Some of the most regularly used descriptors are 2D "fingerprints" [1]. They are binary strings. Because binary representations are the natural currency of computers, so the molecules can be manipulated and compared very rapidly.

The ability to rank molecules according to their effectiveness in some domains (e.g. pesticide and drug discovery) is important, owing to the cost of synthesizing and testing chemical compounds. VS seeks to do this computationally with

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 297–308, 2013.

potential savings of millions of pounds and large profits associated with reduced time to market.

In VS tasks, the problems arise when 2D fingerprints are used as descriptors, described as follows:

1. The fingerprints are high-dimensional containing of $\mathcal{O}\left(10^3\right)$ elements. They are sparse because they only contains a handful of non-zero attribute values. Accordingly, the presence of a feature is more important than the absence of a feature.
2. VS suffers strongly from the so-called small-sample-size problem where the number of covariates is comparable to or exceeds the number of samples. Typically a task in VS comprises a training set sample of size $\mathcal{O}\left(10^2\right)$ of known compounds but with fingerprints of dimension $\mathcal{O}\left(10^3\right)$.

The above problems are one of the grand challenges in the field of data mining and knowledge discovery for engineer, computer scientist, and statistician. Developed algorithms are needed to be robust and efficient. Moreover, they should handle very large set of high-dimensional data because of an unprecedented level of complexity in data model. Thus, complexity control is particular relevant and indispensable in VS tasks. This can be done by developing the machine learning algorithm or integrating a suitable dimensional reduction method in the pre-processing step.

Principal component analysis (PCA) is one of the most widely used dimensional reduction method. Different types of PCA has been introduced to suit particular tasks [2–5]. This paper investigates on five different types of PCA namely conventional PCA, Supervised PCA [2], Supervised Probabilistic PCA [3], Binary PCA [4], and Logistic PCA [5] in VS tasks. These methods are integrated into the pre-processing step of VS task in order to eliminate noise, de-correlate inputs, and make the model simpler. Then the compact samples are screened by linear Fisher discriminant analysis (FDA) and kernel Fisher discriminant analysis (kFDA).

The paper is organized as follows. Section 2 briefly describes PCA and its variants used in this paper together with the main reasons behind the selection of these methods. Section 3 presents the experiment setup and performance comparison of all methods used in this paper to VS.

2 Dimensionality Reduction Techniques

PCA is a popularly used statistical method for dimensionality reduction in multivariate data. It has a natural association with Gaussian distributed data. It attempts to select salient features spanning a space of the lowest possible dimension which is close to a given set of points $\mathrm{X} = [\boldsymbol{x}, \ldots, \boldsymbol{x}_N]^\mathsf{T}$, where $\boldsymbol{x}$ is an m-dimensional vector. Some features in the given set merely constitute noise. PCA provides a new set of points, $\tilde{\mathrm{X}} = [\tilde{\boldsymbol{x}}, \ldots, \tilde{\boldsymbol{x}}_N]^\mathsf{T}$, where $\tilde{\boldsymbol{x}}$ is a n-dimensional vector ($n < m$), which is de-noised. The variation in the dataset can be explained by using just a small number of principal components. Moreover, the

principal components are also much more convenient for graphical data display and analysis.

Bair *et al.* (2006) introduced a technique called Supervised PCA (SPCA) [2]. This uses a subset of the features that are selected based on their association with the outcome while conventional PCA does not take into account available class membership information. The features are selected when the absolute value of standardized regression coefficient is greater than some threshold. The threshold can be identified by using cross validation. SPCA predicts the output by using linear regression model. Their study of gene micro arrays shows that the conventional PCA is not consistent as the sample size and number of features grow, whereas SPCA is consistent.

PCA can be derived from a probabilistic model for the observed data, therefore, Probabilistic PCA (PPCA) has been proposed [6]. The advantage of PPCA is that the projections can be obtained when some data values are missing. Yu *et al.* (2006) believes that there are a covariance between input and output matrix [3]. Hence, they introduced the Supervised PPCA (SPPCA) to model this covariance. It is an extended version of PPCA. The method is derived through an efficient EM learning algorithm.

Regarding the assumption that $\boldsymbol{x}$ is a Gaussian random distribution, Collins *et al.* (2001) argue that this may be improper for data that is not real-valued, such as binary or integer-valued data, or nonnegative data [7]. In fact, the Gaussian assumption suits only real-valued data while the Poisson and the Bernoulli distributions are suited to integer and binary data, respectively. Collins *et al.* (2001) proposed a generalized PCA for the exponential distribution family which is suitable for various types of data [7]. Their results illustrate that this generalized PCA is more robust than the conventional PCA by investigating two synthetic problems. The generalized model's relationship to PCA is analogous to the relationship between logistic and linear regression [8]. In an iterative algorithm for the generalization of PCA proposed by Collins *et al.* (2001) [7], the optimizations required in each iteration do not have a simple closed form for the Logistic PCA (LPCA) case. Hence, a new LPCA which is suitable for dimensionality reduction of binary data is proposed [5]. It is derived from an alternative least squares method and generalized linear coefficient of LPCA. The algorithm is guaranteed at each iteration to improve the LPCA log-likelihood. LPCA is applied on four real world applications and compared with conventional PCA [5]. The experimental results show that LPCA is better suited to reconstruction of binary data than conventional PCA.

Many statistical techniques have been introduced for analysing the binary data e.g. latent structure analysis, multiple correspondence analysis, etc. Thus, PCA particularly for binary data is again developed [4]. It is called Binary PCA (BPCA), which is a combination of latent structure analysis, multiple correspondence analysis, and PCA. The method was first applied to roll-call analysis [4]. It proposes to reduce the dimensionality of binary data by minimizing the distance between the binary matrix and a precise functional form using a majorization algorithm. Hence, it is an EM-like algorithm. Two cases to specify the function

are proposed which are logistic and probit functions. Here, the logistic function is selected in order to enable us to compare directly with LPCA.

In VS, as the length of the binary fingerprint, m, in VS is large – typically in the range 100 to 1000 or more, we need an algorithm which can reduce the size of fingerprint in order to improve the computational time and also the active molecules retrieved rate.

In order to improve the performance of VS, four variants of PCA with potential in VS: SPCA, SPPCA, BPCA, and LPCA are used and compared together. This is done by integrating these methods into the pre-processing step of VS. Then the reduced dimensional data is screened by FDA and kFDA. FDA plays a central role in pattern recognition. It seeks a linear projection that maximizes the separation between data belonging to two classes while minimizing the separation between those of the same class.

The variants of PCA are compared together with the conventional PCA. A reason why these works are repeated is that, none of these methods are compared together with high dimensional and binary-valued data elsewhere. Hence, they are compared together in the experimental framework of VS in order to find which method is the most suitable in VS tasks. These methods are expected to improve the performance of VS rather than using the conventional PCA in pre-processing step. The two main reasons behind the selection of these methods are as follows:

1. Supervised learning – SPCA and SPPCA: They can improve the performance as they take the output into account rather than only input.
2. Logistic function based – BPCA and LPCA: The most useful analogue for binary-valued data of linear model for normally distributed data is provided by logistic function [9].

3 Application to Virtual Screening

Here the MDL Drug Data Report (MDDR) database are used [10]. MDDR database is a set of 102,514 known drugs and biologically relevant molecules collected from patent literature, journals, meetings and congresses. The database is represented by ECFP_4 fingerprints – 1,024-dimensional vectors. ECFP_4 fingerprints uses a circular substructure approach. Each atom is represented by a string of integers obtained by an adaptation of the Morgan algorithm [11]. ECFP stands for extend connectivity fingerprint. The ECFP fragments encode atomic type and charge. "4"denotes the diameter of the circular substructure.

The 11 selected activity classes shown in Tables 1 were selected to reflect typical drug discovery projects for pharmaceutical companies. The mean self-similarity provides a measure of the homogeneity – the degree to which the molecules in the database are alike – of each of the activity classes and a useful way to compare design spreads and coverage for individual activity classes. If there is a high degree of structural similarity between the compounds of an activity class then they will be easier to identify/retrieve in VS. The mean of the similarity of each compound with every other compound in the class, calculated

with the ECFP_4 fingerprint using the Jaccard/Tanimoto similarity coefficient is shown in Table 1. The entries are ranked in decreasing order of homogeneity. All five methods of PCA in the previous section will be integrated in a pre-

Table 1. Self-similarity of the 11 activity classes calculated with the ECFP_4 fingerprint using the Jaccard/Tanimoto similarity coefficient

Activity Class	Class Index	Mean	S.D.	Number of Actives
Renin Inhibitors	1	0.337	0.105	1130
Angiotensin II AT1 Antagonists	2	0.269	0.100	943
HIV Protease Inhibitor	3	0.226	0.101	750
Thrombin Inhibitor	4	0.212	0.098	803
Substance P Antagonists	5	0.179	0.082	1246
5HT3 Antagonists	6	0.175	0.090	752
D2 Antagonists	7	0.173	0.089	395
5HT1A Agonists	8	0.166	0.086	827
5HT Reuptake Inhibitors	9	0.153	0.092	359
Protein Kinase C Inhibitor	10	0.141	0.103	453
Cyclo-oxygenase Inhibitor	11	0.130	0.073	636

processing step in order to reduce the dimensional representations of MDDR database. It should be noted that only SPCA requires a parameter selection (among all methods in previous section). Five-fold cross validation is then applied to SPCA in order to find the optimal parameter on the basis of *sum of active rank position*. Then FDA or kFDA are used as machine learning methods to screen the database. New data are ranked on the predicted output value from most positive (most likely to be active) to most negative. The predicted output value is equal to distance from decision boundary. Here three situations are examined:

1. PCA, SPCA, and SPPCA are compared together in different sizes of reduced dimensional form from 1-D to the required dimension representing 90% of the variance in the original dataset. Then FDA is used to screen the database.
2. All five methods are compared together with only the reduced dimensions of {1,2,4,8} because LPCA and BPCA requires high computational time to run such a large database. Then FDA is used as a classifier.
3. PCA, SPPCA, LPCA, and BPCA are compared together with only reduced dimensional of {1,2,4,8}. kFDA is used as a nonlinear classifier to screen the database. A reason that SPCA is not considered in this situation is that, linear discriminant function is inherent in SPCA.

The experiment was run five times with different random data splits. The number of training samples is equal to 20% of the number of active molecules in the database. The remaining samples are used as a testing set. There are 11 activity classes, therefore, there are 11 classifiers. It is usual in chemoinformatics applications to report the percentage of the maximum possible number of active compounds ranked in the top 5% of the ranked database. All experiments are carried out using the Matlab environment [12]. BPCA and LPCA algorithms are available to download at [3, 5], respectively.

3.1 Comparison of PCA, SPCA, and SPPCA

Comparing PCA, SPCA, and SPPCA together, the best method in the pre-processing step is SPCA when a small number of principal components are used, as shown in Fig. 1 (left). It is better by up to 10.68% and 9.65% than PCA and SPPCA, respectively, when 10% of the variance is accounted for the original data. The percentage of improvement reduces when the percentage of the variance of the original data is increased. SPCA is worse than the other two methods when more than 70% of the variance is accounted for, but the differences are very small. Figure 1 (left) also shows that SPPCA is also better than PCA at 10-60% of the variance.

Figure 1 (right) shows the relative improvement or worsening of PCA and SPPCA with respect to SPCA when 10% of the variance is accounted for in each activity class. The entries are ranked in decreasing order of homogeneity. We can see that SPCA is competitive to PCA and SPPCA in the first two activity classes which are the most homogeneous. On the other hand, SPCA achieves the highest improvement in activity class 11 which is the most heterogeneous activity class. The overall picture is much the same for other percentage of the variance accounted for.

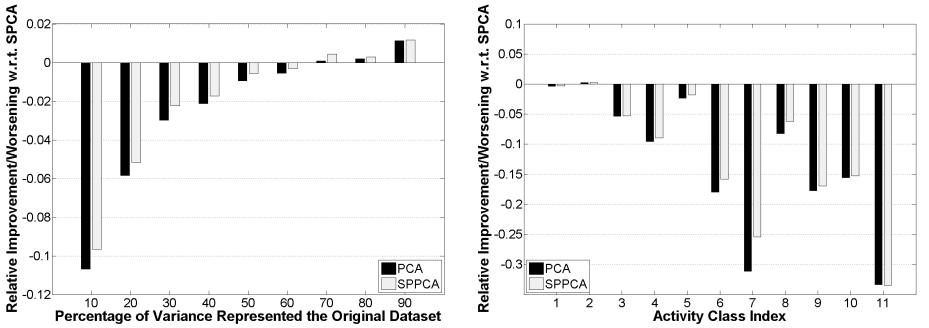


Fig. 1. Relative improvement/worsening of PCA and SPPCA with respect to SPCA based on active molecules retrieved rate, and average across five runs. Left sub-figure shows the average across 11 activity classes at 10-90% of the variance of the original data. Right sub-figure shows the average when 10% variance of the original data is taken into account for each activity class.

Figure 2 shows average proportion of maximum possible active molecules retrieved by a combination of each dimensionality reduction method and linear FDA in class 1 (left) and 11 (right). PCA, SPCA, and SPPCA are compared together for different sizes of reduced dimension from 1-D to the dimension required to represent 90% of the variance in the original dataset. SPCA is better than the other two methods in class 11 (the most heterogeneous) when a small number of principal components are accounted for. SPCA is competitive to the other two methods in class 1 (the most homogeneous). As we can see, the graphs

of SPCA are fluctuated – when the number of represented dimensions increases – but the differences are very small. It should be noted that SPPCA gives almost the same level of performance as PCA does in all reduced dimensional forms.

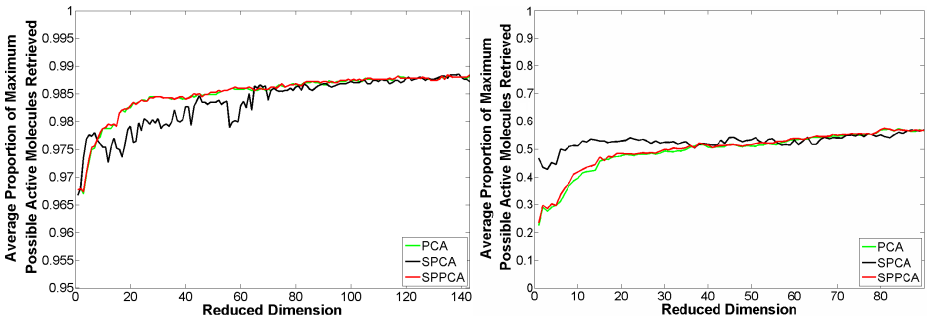


Fig. 2. Average proportion of maximum possible active molecules retrieved by a combination of each dimensionality reduction method and linear FDA in activity class 1 (*left*) and 11 (*right*). PCA, SPCA, and SPPCA are compared together for different sizes of reduced dimension from 1-D to the dimension required to represent 90% of the variance in the original dataset.

3.2 Comparison of All Five Methods

In this subsection, two addition methods, LPCA, and BPCA, are included in the evaluation. Because LPCA and BPCA require high computational time as they are iterative scheme, therefore we consider only reduced dimensional of {1,2,4,8}. Table 2 shows the percentage of active molecules retrieved by a combination of each dimensionality reduction method and linear FDA in a particular reduced dimension. SPCA still yields the best accuracy among all five methods at 70.4% (average across all activity classes and all reduced dimensions) and followed by SPPCA, BPCA, LPCA and PCA on average at 61.8%, 61.7%, 61.1% and 61%. SPCA also achieves the best in all reduced dimensions (average across all activity classes). Its performance is competitive to other methods in the two most homogeneous classes but it considerably overcomes the other methods in other classes. As expected, the performance of each method is improved when more principal components are taken into account.

However, there is a degree of variation in the performance of the various methods as shown in Table 2. Hence, to evaluate their performance, a statistical test is used. The statistical significance of the level of agreement is tested using the *Kendall Coefficient of Concordance* (W) [13] which is used to measure the degree of agreement between the rankings assigned to N objects (here, the 5 different methods) by each of k judges (here, the 11 activity classes). The ranks in Table 3 are assigned according to Table 2 (in two decimal places). When tied ranks occur (same percentage of active molecules retrieved), the ranks are each assigned by the average of the ranks which would have been assigned had no

Table 2. Comparison of PCA (M_1), SPCA (M_2), SPPCA (M_3), BPCA (M_4), LPCA (M_5) in 11 activity classes of MDDR database with reduced dimensional of {1,2,4,8}. Each element represents the mean percentage of active molecules (average across five runs) retrieved in top 5% of ranked database by FDA. The results are reported to one decimal place.

Dim.	1					2					4					8				
Idx	M_1	M_2	M_3	M_4	M_5	M_1	M_2	M_3	M_4	M_5	M_1	M_2	M_3	M_4	M_5	M_1	M_2	M_3	M_4	M_5
1	96.8	96.7	96.8	96.4	94.2	96.8	96.8	96.8	94.9	92.7	**97.0**	97.7	97.1	94.7	92.5	**97.7**	**97.7**	**97.7**	95.7	93.5
2	97.4	97.6	97.4	94.7	93.5	97.6	97.6	97.6	90.6	89.4	97.9	97.6	97.9	89.6	88.4	**98.0**	**98.0**	**98.0**	93.5	92.3
3	72.8	72.8	73.1	72.7	71.6	73.6	73.4	73.6	71.3	70.5	74.3	77.2	74.4	74.7	73.5	76.1	**81.7**	76.2	77.8	76.6
4	56.3	70.8	57.4	61.9	60.6	65.7	76.5	66.3	62.7	61.4	71.3	80.0	71.9	66.7	65.3	76.0	**85.7**	76.1	76.6	75.0
5	52.6	72.3	56.0	50.0	49.5	54.2	76.1	57.1	51.7	51.5	67.5	80.6	70.4	60.9	60.2	80.3	**82.1**	80.8	63.8	63.1
6	44.1	74.1	46.1	51.7	52.6	52.6	72.6	54.0	58.8	57.8	54.6	77.6	57.0	62.3	61.3	71.4	**79.4**	72.3	68.1	67.0
7	42.5	55.8	42.9	46.4	45.8	39.3	58.8	40.3	48.2	47.7	50.6	**60.9**	51.7	51.1	51.0	56.2	59.8	56.9	52.4	54.9
8	52.0	59.3	53.2	51.0	50.4	53.9	68.0	54.7	59.8	59.1	58.3	69.4	58.1	66.1	65.3	67.8	**73.7**	71.1	68.4	67.6
9	25.1	44.8	25.6	35.4	36.4	37.7	**50.8**	38.5	37.7	37.3	44.1	45.7	44.5	44.3	44.5	49.0	50.4	49.7	46.8	47.8
10	34.2	44.5	34.9	44.7	44.0	40.7	47.7	44.4	45.3	44.6	44.3	52.4	44.4	46.8	46.1	46.8	**56.2**	47.1	44.5	45.3
11	22.6	46.8	23.6	23.5	27.8	29.0	43.6	29.7	36.6	36.0	29.0	45.2	30.2	44.9	44.1	37.0	**50.9**	37.9	41.3	40.6
Avg.	54.2	66.9	55.2	57.1	56.9	58.3	69.2	59.4	59.8	58.9	62.6	71.3	63.4	63.8	62.9	68.8	**74.1**	69.4	66.3	65.8
Avg.	58.1					61.1					64.8					**68.9**				

ties occurred [13]. Assume that $\overline{R_i}$ is the average of the ranks assigned to the i^{th} object, then the Kendall coefficient of concordance is

$$W = \frac{12 \sum \overline{R_i}^2 - 3N(N+1)^2}{N(N^2-1) - \frac{\sum T_j}{k}} \tag{1}$$

T_j is a correction factor which can be calculated from,

$$T_j = \sum_{i=1}^{g_j} (t_i^3 - t_i) \tag{2}$$

where t_i is the number of tied ranks in the i^{th} grouping of ties, and g_i is the number of groups of ties in the j^{th} setup ranks.

The significance of the computed value of W can be obtained from the table of critical values for $N \leq 7$ [13] or from a table of the chi-square distribution with $N-1$ degrees of freedom for $N > 7$. The chi-square can be calculated from

$$k(N-1)W \sim \chi^2_{N-1} \tag{3}$$

The computed value of W for each evaluated size of reduced dimensional form is shown in Table 4. These values are significant at $p < 0.01$. Given that a significant level of agreement between various rankings of the same set of objects has been established, the best overall order of the N objects is obtained by taking the mean ranks, $\overline{R_i}$ [13], which suggests the ordering in Table 4.

The performance of SPCA are also compared together with FDA as shown in Table 5. Accuracy is reported along with the percentage of principal component used (below). SPCA is competitive to FDA and displays the best accuracy in 7/11 classes. However, FDA uses all the provided information (1,024-D vector) while SPCA uses only 10.1% of the provided information.

Table 3. Ranks assigned to PCA (M_1), SPCA (M_2), SPPCA (M_3), BPCA (M_4), LPCA (M_5) for Table 2 (in two decimal places)

Dim.	1					2					4					8				
Idx	M_1	M_2	M_3	M_4	M_5	M_1	M_2	M_3	M_4	M_5	M_1	M_2	M_3	M_4	M_5	M_1	M_2	M_3	M_4	M_5
1	1.5	3	1.5	4	5	2.5	1	2.5	4	5	3	1	2	4	5	1.5	3	1.5	4	5
2	3	1	2	4	5	2	3	1	4	5	1.5	3	1.5	4	5	2.5	1	2.5	4	5
3	2.5	2.5	1	4	5	2	3	1	4	5	4	1	3	2	5	5	1	4	2	3
4	5	1	4	2	3	3	1	2	4	5	3	1	2	4	5	4	1	3	2	5
5	3	1	2	4	5	3	1	2	4	5	3	1	2	4	5	3	1	2	4	5
6	5	1	4	3	2	5	1	4	2	3	5	1	4	2	3	3	1	2	4	5
7	5	1	4	2	3	5	1	4	2	3	5	1	2	3	4	3	1	2	5	4
8	3	1	2	4	5	5	1	4	2	3	4	1	5	2	3	4	1	2	3	5
9	5	1	4	3	2	3.5	1	2	3.5	5	5	1	2.5	4	2.5	3	1	2	5	4
10	5	2	4	1	3	5	1	4	2	3	5	1	4	2	3	3	1	2	5	4
11	5	1	3	4	2	5	1	4	2	3	5	1	4	2	3	5	1	4	2	3
R_i	43.0	15.5	31.5	35.0	40.0	41.0	15.0	30.5	33.5	45.0	43.5	13.0	32.0	33.0	43.5	37.0	13.0	27.0	40.0	48.0

Table 4. Computed W for each evaluated size of reduced dimensional form and overall order of all methods with FDA

Dimension	W	Ordering				
1	0.3849	SPCA	> SPPCA	> BPCA	> LPCA	> PCA
2	0.4491	SPCA	> SPPCA	> BPCA	> PCA	> LPCA
4	0.5183	SPCA	> SPPCA	> BPCA	> PCA	> LPCA
8	0.6055	SPCA	> SPPCA	> PCA	> BPCA	> LPCA

Table 5. Comparison of maximum percentage actives retrieved in top 5% of ranked database using SPCA and percentage of principal component used (below)

Idx	1	2	3	4	5	6	7	8	9	10	11	Average
SPCA	98.8	**98.5**	90.9	**91.9**	**87.6**	**86.9**	66.5	**77.7**	**64.4**	63.7	**56.6**	**80.3**
	14.1	12.5	10.0	10.8	16.0	10.2	5.7	11.0	5.2	6.5	9.0	**10.1**
FDA	**98.9**	**98.5**	**91.7**	91.6	85.5	84.7	**68.7**	75.6	61.5	**71.4**	55.9	**80.3**
						100.0						100.0

The performance of BPCA and LPCA are compared with PCA as shown in Fig. 3 (left). The performance of BPCA and LPCA are slightly worse than PCA in 5/11 cases but they perform better relative to PCA 5% or greater on average in the six most heterogeneous classes. Moreover, BPCA and LPCA provide evidence for the superiority of the algorithm on the most heterogeneous class by 25% better relative to PCA.

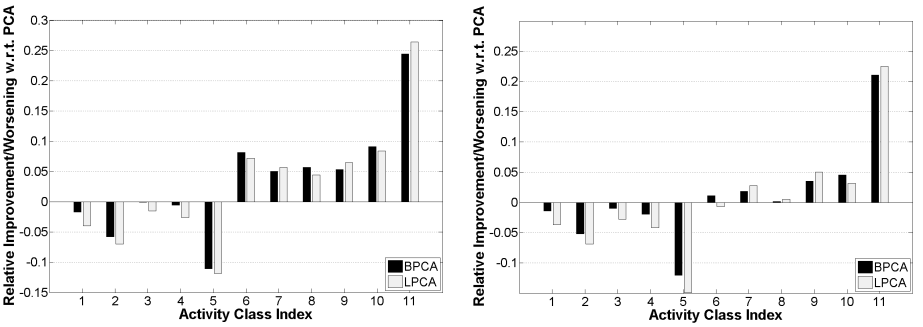


Fig. 3. Relative improvement/worsening of BPCA and LPCA with respect to PCA based on active molecules retrieved rate by linear FDA (left) and kFDA (right). (average across five runs, 11 activity classes, and all evaluated size of reduced dimensional form)

Table 6. Ranks assigned to PCA (M_1), SPPCA (M_3), BPCA (M_4), LPCA (M_5) according to the mean percentage of active molecules (average across five runs) with reduced dimensional of {1,2,4,8} retrieved in top 5% of ranked database by kFDA.

Dim.	1				2				4				8			
Idx	M_1	M_3	M_4	M_5	M_1	M_3	M_4	M_5	M_1	M_3	M_4	M_5	M_1	M_3	M_4	M_5
1	1.5	1.5	3	4	2	1	3	4	2	1	3	4	2	1	3	4
2	2	1	3	4	2	1	3	4	2	1	3	4	2	1	3	4
3	2	1	3	4	2	1	3	4	2	3	1	4	2	1	3	4
4	4	3	1	2	2	1	3	4	1	2	3	4	2	1	3	4
5	2	1	3	4	2	1	3	4	2	1	3	4	2	1	3	4
6	4	3	2	1	4	3	1	2	2	1	3	4	2	1	3	4
7	4	3	1	2	4	3	1	2	2	1	3	4	2	1	4	3
8	3	2	4	1	4	3	1	2	1.5	1.5	3	4	2	1	3	4
9	4	3	2	1	3	1	2	4	2	1	4	3	2	1	4	3
10	4	3	1	2	4	1	2	3	2	3	1	4	1	2	3	4
11	4	2	3	1	4	3	1	2	4	3	1	2	4	3	1	2
R_i	34.5	23.5	26.0	26.0	33.0	19.0	23.0	35.0	22.5	18.5	28.0	41.0	23.0	14.0	33.0	40.0

3.3 Comparison of PCA, SPPCA, BPCA, and LPCA Using kFDA

This subsection compares four different types of PCA. They are integrated into the pre-processing step of VS, then the data is screened by kFDA. In the experiments, a radial basis function is used as the kernel function. Five-fold CV enabled us to obtain the optimal model (kernel and regularization parameter).

The overall picture is much the same as the use of linear classifier. SPPCA yields the best accuracy among all four methods at 63.51% and followed by PCA, BPCA, and LPCA at 62.60%, 62.15%, and 61.37%, respectively. Again, there is a degree of variation in the performance of four methods. Therefore, we used Kendall Coefficient of Concordance (W) to test the significance of the level of agreement. The ranks in Table 6 are assigned according to the mean percentage of active molecules (average across five runs) retrieved in top 5% of ranked database by kFDA. The computed value of W for each evaluated size of reduced dimensional form is shown in Table 7. Only reduced dimensions of 2,4,8

Table 7. Computed W for each evaluated size of reduced dimensional form and overall order of all methods with kFDA

Dimension	W	Ordering			
1	0.1159	Not Significant			
2	0.2959	SPPCA	> BPCA	> PCA	> LPCA
4	0.4812	SPPCA	> PCA	> BPCA	> LPCA
8	0.6430	SPPCA	> PCA	> BPCA	> LPCA

Table 8. Comparison of active retrieved rate by linear FDA and kFDA

Classifier	PCA	SPPCA	BPCA	LPCA	Mean
FDA	60.96	61.84	61.74	61.14	61.42
kFDA	**62.60**	**63.51**	**62.15**	**61.37**	**62.41**

are significant at $p < 0.05$. Hence, the best overall order of the four variants of PCA for each evaluated size of reduced dimensional form is shown in Table 7.

As before, BPCA and LPCA display improvement on accuracy with respect to PCA in the six most heterogeneous classes but are worse in 5/11 cases as shown in Fig. 3 (right). The performances of linear and nonlinear classifiers are compared together as shown in Table 8. The results show an improvement on accuracy (1% on average) in the four variants of PCA in conjunction of kFDA.

4 Conclusion

The relevant dimensionality reduction methods have been compared toward application in VS. SPCA is the best method when a small number of principal components is used, followed by SPPCA. These two methods are supervised methods. This validates the hypothesis that supervised learning can improve performance. The experiments also provide evidence for the superiority of the SPCA on the heterogeneous activity classes. SPCA is found to be competitive to FDA but FDA uses all the provided information. This is important since many commercial databases contain $\mathcal{O}\left(10^6\right)$ samples and speed of recall can be an issue.

BPCA and LPCA outperform PCA in the heterogeneous classes. On the other hand, their performance is worse than PCA in homogeneous classes. The reason that they degenerate the solutions is that, they face the problem of over-fitting [4, 14]. However, it is not worth using them in this problem because they consume high computational time, and speed of recall is important in VS tasks. The projection of data to the low-dimensional space by BPCA and LPCA must be done iteratively while this can be computed directly by conventional PCA.

Linear FDA and kFDA are used to rank the low-dimensional space samples in database. The experiments show an improvement in performance by using the nonlinear classifier. Hence, it is worthwhile to embed the nonlinear discriminant function into SPCA instead of linear discriminant function.

References

1. Leach, A.R., Gillet, V.J.: An Introduction to Chemoinformatics. Kluwer Academic Publishers, Dordrecht (2003)
2. Bair, E., Hastie, T., Paul, D., Tibshirani, R.: Prediction by Supervised Principal Components. J. Am. Statist. Assoc. 101, 119–137 (2006)
3. Yu, S., Yu, K., Tresp, V., Kriegel, H.P., Wu, M.: Supervised Probabilistic Principal Component Analysis. In: Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining, pp. 464–473. ACM Press, New York (2006)
4. de Leeuw, J.: Principal Component Analysis of Binary Data by Iterated Singular Value Decomposition. Comput. Stat. Data An. 50(1), 21–39 (2006)
5. Schein, A.I., Saul, L.K., Ungar, L.H.: A Generalized Linear Model for Principal Component Analysis of Binary Data. In: Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics (2003)
6. Tipping, M.E., Bishop, C.M.: Probabilistic Principal Component Analysis. J. R. Stat. Soc. Ser. B 61, 611–622 (1999)
7. Collins, M., Dasgupta, S., Schapire, R.E.: A Generalization of Principal Components Analysis to the Exponential Family. In: Advances in Neural Information Processing Systems, pp. 617–624 (2001)
8. Mccullagh, P., Nelder, J.A.: Generalized Linear Models. Chapman & Hall, London (1989)
9. Cox, D.R., Snell, E.J.: Analysis of Binary Data, 2nd edn. Chapman & Hall, London (1989)
10. Tinterwordspacing MDL Information Systems Inc.: The MDL drug data report database (2006), `http://www.mdli.com`
11. Morgan, H.L.: The Generation of a Unique Machine Description for Chemical Structure – A Technique Developed at Chemical Abstracts Service. J. Chem. Doc. 5, 107–113 (1965)
12. MathWorks Inc.: Matlab Version 7.10 (2010),`http://www.mathworks.com`
13. Siegel, S., Castellian, N.J.: Nonparametric Statistics for the Behavioral Sciences, 2nd edn. McGraw-Hill, Singapore (1988)
14. Kabán, A., Bingham, E., Hirsimäki, T.: Learning to Read Between the Lines: The Aspect Bernoulli Model. In: Proceedings of 4th SIAM International Conference on Data Mining, pp. 462–466. SIAM, Florida (2004)

Improved X-ray Edge Detection Based on Background Extraction Algorithm

Jakub Romanowski, Tomasz Nowak,
Patryk Najgebauer, and Sławomir Litwiński

Institute of Computational Intelligence, Częstochowa University of Technology
al. Armii Krajowej 36, 42-200 Częstochowa, Poland
{jakub.romanowski,tomasz.nowak,patryk.najgebauer,
slawomir.litwinski}@iisi.pcz.pl
http://iisi.pcz.pl

Abstract. Digital X-ray imaging is a source of generous information about health of patient bones. One of major obstacles in computer analysis of digital X-ray images is the presence of bone tissue and soft-tissue areas. It has a negative impact on the quality of bone edge detection or detection of bones area on X-ray images. The main goal is to create an efficient method of edge detection which performs efficiently on properly prepared digital X-ray images. This paper describes a new method of background removal from X-ray images where the background is in the form of soft-tissue. The aim of this is to prepare the image to the next step of processing. We also present a new approach to edge detection of bones on X-ray images. Performance of the proposed method is achieved by eliminating unnecessary areas of the image which are not bone tissue and which are not the main region of interest. Additionally, the presented method of edge detection is partly based on known algorithm named Integral Image and specific edge detection filter, what allow to achieve the desired objectives.

Keywords: X-ray images, edge detection, background substraction, medical image processing.

1 Introduction

There exist many computer systems and algorithms for analysing medical images [11][12][21] and data [3][5][7][6][8][15][16][19][20]. X-ray images in a digital form can be seen as some kind of signal which contains large amount of important information, but also a lot of noise or other unnecessary information. The occurrence of unwanted parts of the signal causes problems in image processing which results in obtaining incorrect results. The most important advantage of the new algorithms presented in the paper is the ability to distinguish an image area associated only with the bone tissue. Areas of image which contains soft-tissue such as skin or fat introduce a lot of noise. Many methods of edge detection are applied directly to the unprocessed image what is the reason that bone edges are equally important and clearly marked as, for example layered skin. Significant bending of skin or fat could be seen as clear as the edges of the bone.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 309–319, 2013.

In many cases humans are able to recognize these two kinds of tissues, but there are images where it is not so easy due to their high brightness. The problem becomes more significant when we look at it from the point of computer image analysis, where there is not any intuition, analytical thinking or inference. The method developed here allows to some extent to reproduce the process of human analysis of X-ray images, to distinguish different types of tissue and to separate them from each other in a way appropriate for the edge detection stage of image processing. The method of obtaining bone areas from X-ray images gives a lot opportunities to develop information systems that operate on X-ray image comparison algorithms. For proper functioning, these kind of systems require thoroughly extracted bone areas, which in extracted form could be processed or compared by another algorithms of image processing. The comparison of two or more objects could be much more accurate, more correct and faster in comparison to the areas of bone detected by previously used methods. Conviction to develop new methods is caused by differences in the method of X-ray imaging in hospitals. Already developed methods give clear images of soft-tissue or bone tissue. The problem of these methods is that they require two X-ray exposures with different levels of radiation. This kind of method is so called DEXA (Dual Energy X-ray Absorptiometry) [1][22][9]. The reason to develop a new method is that a lot of medical centers perform only single X-ray exposure, which does not allow to use such an image in DEXA methods. Available edge detection algorithms allow for detecting edges of bones but they also detect soft-tissue edges [18]. Modification of existing filters of edge detection algorithms to use them to digital X-ray processing can bring appropriate results. A combination of two solutions should give satisfactory results.

1.1 Background Removal – Soft-Tissue Extraction

DEXA (Dual-Energy X-ray Absorptiometry) is a widely used method used to extract soft tissue or bone tissue from X-ray images. It produces two different images containing each of them. The algorithm can work only on two X-ray images taken with different radiation levels. This method could be used for X-ray [1][22] as well as for CT imaging. This method gives excellent results, but the problem is that two images are required. When the density of bones, soft-tissue and level of radiation are known it is possible to create two different images, one containing only soft-tissue and second which containing only bones. In many medical centers this method is impossible to use because usually only single X-ray image is taken with only one level of radiation. It is the main reason to create an efficient method which will be able to extract background (soft-tissue) from digital X-ray images and it will utilize only one image. The aim is to speed up the detection of bone area and increase the accuracy of the bone location in the image. There are also edge detection methods which process images which contain bones and soft-tissue, but after edge detection stage there are visible edges of bones and edges of soft-tissue caused by heaps of fat and skin and often their brightness is at the same level as edges of bones. To these kind of edges sets which contain 50% of redundant data, an operation is performed to

fit the pre-specified pattern to extracted earlier edges set from the X-ray image [17][14]. Fitting edges to the pattern is performed also on redundant edges, so fitting operation is much more longer than it could be. It takes a lot of time and it is computationally complex. This is another reason in favor to the creation of new solution that would reduce redundant computations associated with the soft-tissue edges set detected on digital of X-ray.

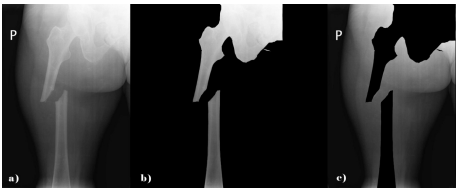


Fig. 1. Figure presents: (a) original of X-ray, (b) bone tissue, (c)soft-tissue named as background (skin, fat, muscles)

1.2 Edge Detection

Edge detection methods used so far to detect bone edges are mainly based on common operations such as second derivative Laplace operator [18][13], Roberts operator, Sobel operator, Prewitt operator or Canny edge operator [18][10]. Their important common feature is a type of filter used to detect the edges. These filters process individual pixels used for edge detection in the X and Y axis. They sum values of pixels from specific areas of filter masks and the resulting value determines absence or presence of edge. For example, Laplace operator is described as the second derivative of the pixels values obtained from different values of the pixels included in the masks. Laplace operator could be described as follows [13]

$$\delta_x^2 = \nabla \Delta f(x,y) \tag{1}$$

$$= \Delta(f(x+1,y) - f(x,y)) \tag{2}$$

$$= (f(x+1,y) - f(x,y)) - f(x,y) - (f(x,y) - f(x-1,y)) \tag{3}$$

$$= f(x+1,y) - 2f(x,y) + f(x-1,y) \tag{4}$$

The matrix form of the filter is presented as follows [18]:

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{5}$$

Second derivative in x-direction Second derivative in y-direction

These operations are time-consuming because filters are moving through the image with one pixel steps and in this way areas suspected as the edges are designated. So small moving step of filter and so small examination areas inform of one pixel, allows to detect only very significant parts of edges on any image also on X-ray. With this kind of edge detection it is impossible to determine blurred objects edges, that is, the edges in some kind of gradient on the image. These reasons are for using another method which allow for fast edge detection with the largest possible correctness. Using a different type of the filter we propose a method where areas of filter masks in the form of one single pixel are replaced by pixel sets represented by the form of so called integral image [2].

1.3 Integral Image

Integral image is the name of the algorithm used to calculate average pixel values in selected regions of the image. This method is based on a rectangular areas of image, where with specific pattern average value of region of pixels is counted. This algorithm is based on high-speed calculations of rectangular areas of pixels, which in computation complexity are much more efficient and less overload in comparison with the calculation of the average value calculated with use of standard methods [23]. The value of selected pixel specified by position (x, y) consists the sum of pixels above and the left of it. Integral image can be represented as the following pattern [2]:

$$I_{\sum}(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \tag{6}$$

where $I_{\sum}(x)$ is once calculated Integral Image designated by area of rectangle with the beginning at the first pixel of image (point O), and the ending at the point $x = (x, y)$ of original image [2]. An important issue of using this algorithm in our approach is to calculate one-time Integral Images of the entire image. Individual average values of pixels fields we are interested could be calculated by only three addition operations, as shown in Fig 2 [23][2]. Due to the fact that it is aimed at detecting the edges of X-ray images based on modified filters

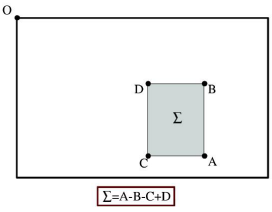


Fig. 2. An example how to determine the average value of pixels in the area of the selected rectangle described by points A, B, C and D based on the values calculated by the integral images [2][4]

where calculation areas size are more than one pixel using of Integral Images is fully justified. Calculation with simple addition operations it is possible to immediately receive average value of filter cells which will be compared. Filter cells are not a sets of single pixels but sets of pixels treated as one value (the average value of pixels in the area). For this reason we used Integral Image algorithm to calculate components of edge detection filters.

2 Block Diagram of the Proposed Method

Edge detection algorithm preceded by removal of the background presented in this paper can be divided into several stages presented in Fig. 3. First stage

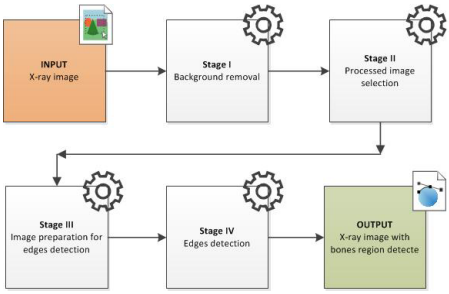


Fig. 3. Block diagram of the proposed methods

of original digital X-ray image processing is background removal (soft-tissues). This stage is intended to get rid of possible largest non-bone area. In order to achieve the intended effect there was developed support software for tests and imaging results of our assumptions. To achieve the background removal, we developed a recursive method for computing the total difference between the reference image and one generated on the basis of a reduced image brightness. A detailed description of the method is presented later in this paper. Image on which bone tissue is visible preferably is undergo another processing operations boiling down to determine the edges of the image. The edge detection method is the last stage of processing X-ray image and its solution is presented in the following sections of the paper. It is worth to mention that the efficiency of edges detection is increased by changing edge detection filter to one based on Integral Image concept.

3 Background Removal Algorithm

Background removal algorithm involving extraction of unwanted parts of the image in the form of soft-tissue such as fat, muscles and skin was designed to prepare the X-ray image to the next stage of edge detection. By using this method it is possible to pre-eliminate edges formed with layering fat, skin or

muscles. The apparent and incorrect edges in this form are the cause of many significant edges detection errors in X-ray image. Their brightness and width is often nearly equivalent to the brightness of the bone area. Depending on the amount of skin, muscles and fat in the image, there could be less or more erroneous readings. An important observation is the fact that X-rays penetrability of bones will always be much smaller than even the plated skin or fat. In result, the brightness of plated soft tissue is always lower than the area of bone tissue. This difference may be invisible to the human eye but numerical values of image pixels show noticeable differences. It is one of the most important properties of X-ray imaging that allows for the use of the proposed background removal methods. During developing this method, bone brightness property was used to determine a specific brightness value as the value of the so-called "protected value". The protected value is some kind of value equivalent to the value of bone brightness in the X-ray image. This value is treated as a non-modifiable during some processing operations of this stage. It is very important in the preparation of special mask which represents soft-tissue area. The range of protected value is included in space of gray scale images brightness and it is from 1 to 256. Background removal method consists of several steps of image processing:

1. On the basis of the original image there is generated a set of additional images with reduced color brightness. Brightness of protected values is not reduced. Software developed for the tests generates a set of images with different protected value for each of them. Basing on this images set the appropriate image with reduced brightness can be selected. Fig. 4 presents

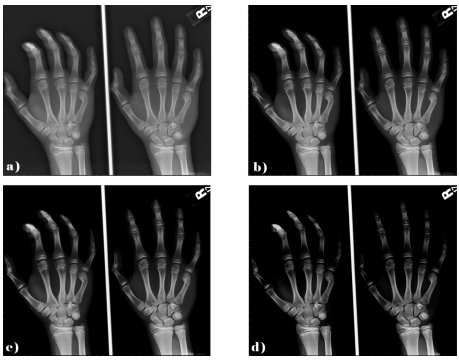


Fig. 4. Part of all images set after removing pixel brightness values without removing protected value. Successively presented images show protected values which are: (a) 105 , (b) 135, (c) 150, (d)180

the results of the algorithm which are largely devoid of soft tissue but it is an image with ideal exposure of light and it have not any areas with highest brightness than other parts of image. It might be called the ideal case of X-ray. For this reason, pre-processing operation gives the image with

satisfactory results. Fig. 5 shows another case where the pre-processing stage area appropriate and not sufficient. It is necessary to further process the X-ray image. Fig. 5 shows how a significant overexposure is affecting the

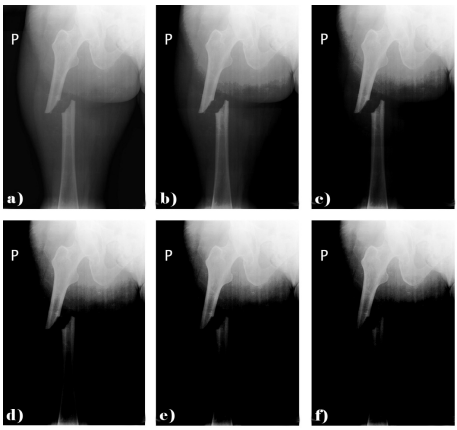


Fig. 5. Part of all image set after removing pixel brightness values without removing protected value. Presented images show protected values of, respectively: (a) 105, (b) 120, (c) 135, (d) 150, (e) 165, (f) 180.

quality of image processing results. To get most interesting area of the image (vertical bone area), it is necessary to create a differential mask. Fig. 5 presents an extreme case of X-ray with higher brightness in some area of image but in practice of medical centers this kind of X-ray are still repetitive and quite frequent. It is the reason that further processing is required in order to extract possible clearest image of bones without soft-tissue. This list of images visible in Fig. 5 is just illustrative to visualize the algorithm results with different protected values. One of images with selected protected value must be chosen to be passed to the next stages of processing.

2. The original not processed image is matched with an image with reduced brightness. On the basis of the total difference in pixel values of these two images soft-tissue pixel mask is generated.
3. After receiving pixel mask, first differential comparison is performed between original image and received mask.
4. Then, next differential comparison are performed but difference is in that the received earlier pixel mask is juxtaposed with the image received from step 3(image received from previous step) of this algorithm.
5. Iterations of differential comparison are performed until image is free up a large amount of soft-tissue areas with maintaining visibility of the area of bones as much as possible.

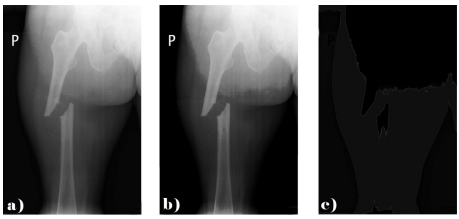


Fig. 6. Pixel mask obtained on the basis of the total difference in the reference image and the image processed using protected value. There are presented (a) original image, (b) image processed with protected value, (c) soft-tissue pixel mask

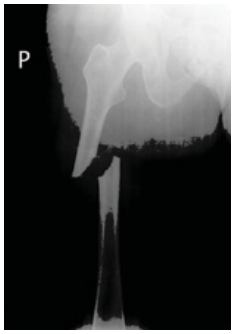


Fig. 7. Result of the iterative pixel mask subtraction

4 Edge Detection Algorithm

Main goal of edge detection algorithm is to create edge detection filter, which cells are based on groups of pixels, not on single pixels values as is in popular edge detection algorithms. Additionally, edge detection with conjunction with background removal algorithm lead to more accurate detection of edges distorted by higher lighting on some areas of digital of X-ray. Values of new filters cells described as groups of pixels are average value of all pixels in this area (filter cell). Average value of filter cell is calculated by Integral Image algorithm. This allows to make quick calculations of the average values of pixels group in filter cells without any need to summing all the pixels of filter cell area and finally divide them by the number of pixels in selected area. With few simple operations it is possible to determine average pixels value for each sub-area of filter in accordance with the rule shown in Fig. 2. Increasing the filter cells size allows to detection changes in edges of larger area in relation to the standard used filters. With this method it is possible to detect the edges which are not possible to detect with popular edge detection algorithms ant their filters with one pixel area cells.

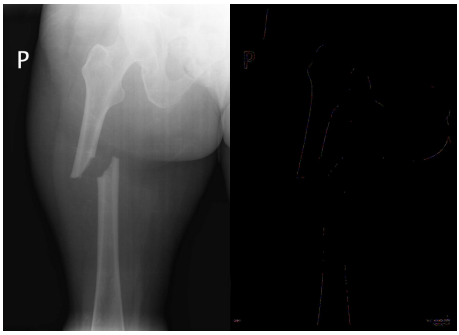


Fig. 8. Results of edge detection without previous background removal processing

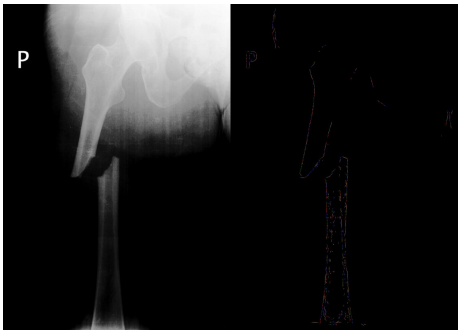


Fig. 9. Results of edge detection with previous background removal processing

The test shows the effect of edge detection with filter cell size 4x4 pixels. The results of application of this filter for image before background removal processing presents Fig 8 and the results of application of this filter in connection with background removal algorithms is shown in Fig. 9.

5 Conclusions

During the test on X-ray images like edge detection may be encountered in images of different quality of performance from those that may be called as ideal upon such even the human eye is not easy to recognize borders between bones and soft-tissue for reason of significantly high lightened regions occurring in images. In case of X-ray images taken correctly there is no issue and background extraction method in connection with proposed in this paper edge detection method works very well. However often there is a second type of images where identification of borders of objects is much more difficult. This paper presents experimental results with stronger exposed images which processing is much more difficult to achieve. Finally the images with clear edges of bones on X-ray. Research proves the efficiency of the proposed algorithms of background

extraction and edges detection. The final result is shown in Fig. 9 in comparison with the Fig. 8 where background extraction method was not used. In Fig. 8 and Fig. 9 significant difference in the quality, continuity and amount of detected edges are visible. The test image selected as example in experiments is very high-lightened in the upper right corner. Edges visible as a result of research show a clearer image of bone and larger number of more consistent edges. The image which is the result of research show pronounced the main bones parts and fractures what is also important information which gives a number of possibilities for X-ray analysis. Methods proposed in this paper are enriched with additional calculations for background removing but quality of the obtained results in form of much more visible edges of bones should be sufficient reason for its use.

Acknowledgments. This work was supported by the Foundation for Polish Science – TEAM project 2010-2014.

References

1. Bayar, A., Sarikaya, S., Keser, S., Ozdolap, S., Tuncay, I., Ege, A.: Regional bone density changes in anterior cruciate ligament dificient kness: A DEXA study. Knee (2008)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. Computer Vision and Image Understanding (2008)
3. Cpałka, K.: On evolutionary designing and learning of flexible neuro-fuzzy structures for nonlinear classification. Nonlinear Analysis Series A: Theory, Methods and Applications 71(12), e1659–e1672 (2009)
4. Evans, C.: Notes on the OpenSURF Library (2009), `http://cs.bris.ac.uk/Publications`
5. Korytkowski, M., Gabryel, M., Rutkowski, L., Drozda, S.: Evolutionary Methods to Create Interpretable Modular System. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 405–413. Springer, Heidelberg (2008)
6. Korytkowski, M., Rutkowski, L., Scherer, R.: On combining backpropagation with boosting. In: IEEE International Joint Conference on Neural Network, Vancouver, Canada, July 16-21, pp. 1274–1277 (2006)
7. Korytkowski, M., Gabryel, M., Rutkowski, L., Drozda, S.: Evolutionary methods to create interpretable modular system. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 405–413. Springer, Heidelberg (2008)
8. Nowicki, R.: On classification with missing data using rough-neuro-fuzzy systems. International Journal of Applied Mathematics and Computer Science 20(1), s. 55-s.67 (2010)
9. Nowicki, R., Pokropińska, A.: Information Criterions Applied to Neuro-Fuzzy Architectures Design. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 332–337. Springer, Heidelberg (2004)
10. Maini, R., Aggarwal, H.: Study and Comparison of Various Image Edge Detection Techniques. International Journal of Image Processing (2009)

11. Ogiela, L., Tadeusiewicz, R., Ogiela, M.R.: Cognitive Analysis in Diagnostic DSS-Type IT Systems. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 962–971. Springer, Heidelberg (2006)
12. Ogiela, M.R., Tadeusiewicz, R.: Artificial Intelligence Structural Imaging Techniques in Visual Pattern Analysis and Medical Data Understanding. Pattern Recognition 36(10), 2441–2452 (2003)
13. Paris, R.B.: The discrete analogue of Laplace's method. Computers & Mathematics with Applications (2011)
14. Pilgram, R., Walch, C., Blauth, M., Jaschke, W., Schubert, R., Kuhn, V.: Knowledge-based femur bone detection in conventional radiographs of the pelvis. Computers in Biology and Medicine (2008)
15. Scherer, R., Rutkowski, L.: Neuro-fuzzy relational classifiers. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 376–380. Springer, Heidelberg (2004)
16. Scherer, R., Rutkowski, L.: A fuzzy relational system with linguistic antecedent certainty factors. In: 6th International Conference on Neural Networks and Soft Computing, Zakopane, Poland, June 11-15. Advances in Soft Computing, pp. 563–569 (2002)
17. Schreiber, J., Schubert, R., Kuhn, V.: Femur Detection in Radiographs using Teplate-based Registration. Bildverarbeitung für die Medizin (2006)
18. Shubhangi, D.C., Raghavendra, S., Chinchansoor, P., Hiremath, S.: Edge Detection of Femur Bones in X-ray images - A comparative study of Edge Detectors. International Journal of Computer Applications (2012)
19. Szarek, A., Korytkowski, M., Rutkowski, L., Scherer, R., Szyprowski, J.: Application of Neural Networks in Assessing Changes around Implant after Total Hip Arthroplasty. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 335–340. Springer, Heidelberg (2012)
20. Szarek, A., Korytkowski, M., Rutkowski, L., Scherer, R., Szyprowski, J.: Forecasting Wear of Head and Acetabulum in Hip Joint Implant. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 341–346. Springer, Heidelberg (2012)
21. Tadeusiewicz, R., Ogiela, L., Ogiela, M.R.: Cognitive Analysis Techniques in Business Planning and Decision Support Systems. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 1027–1039. Springer, Heidelberg (2006)
22. Whitman, G.J., Niklason, L.T., Pandit, M., Oliver, L.C., Atkins, E.H., Kinnard, O., Alexander, A.H., Weiss, M.K., Sunku, K., Schulze, E.S., Greene, R.E.: Dual-Energy Digital Substraction Chest Radiography: Technical Considerations. Curr Problems in Diagnostic Radiology (2002)
23. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. Computer Vision and Pattern Recognition (2001)

Comparison of Time-Frequency Feature Extraction Methods for EEG Signals Classification

Grzegorz Rutkowski[1,⋆], Krzysztof Patan[1], and Paweł Leśniak[2]

[1] Institute of Control and Computation Engineering,
University of Zielona Góra,
ul. Podgórna 50, 65-246 Zielona Góra, Poland
[2] Ward of Neurology and Strokes
Provincial Hospital of Zielona Góra, Poland
k.patan@issi.uz.zgora.pl

Abstract. Biomedical signal analysis can assist the diagnostic process for each neurological dysfunction. At that time the diagnostic tool continues to evolve, which is related to the development of both technological and innovative solutions proposed by scientists. One of possible solutions for automatic EEG analysis is to use modern signal processing tools, which are able to give a time-frequency representation of a signal. This paper presents a comparison of methods such as the discrete wavelet transform, the matching pursuit algorithm, and the S transform for feature extraction and then classification of neurological disorders. The research was carried out using real EEG recordings of epileptic patients as well as healthy subjects.

Keywords: EEG signals, time-frequency representation, classification.

1 Introduction

There are many technique for biomedical signal measurements [14,4,8,2]. One of the most popular methods is to measure the activity of neurons on the surface of the cerebral cortex known as ElectroEncephaloGraphy (EEG). This consists in measuring the neural activity on the surface of the cortex by placing electrodes on the scalp to the standards set for this type of measurement. The analysis of these measurements contributes to the subsequent process of diagnosis accuracy and sets the direction for the appropriate treatment for a particular path of dysfunction. Attempts to search for new solutions in the area of artificial neural networks for complex disorders such as epileptic cases will give the answer to the question of how the human brain works and what processes may be responsible for the occurrence of data dysfunction.

⋆ This work was supported in part by a grant for young scientists and doctoral students of the Faculty of Electrical Engineering, Computer Science and Telecomunications, University of Zielona Góra

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 320–329, 2013.

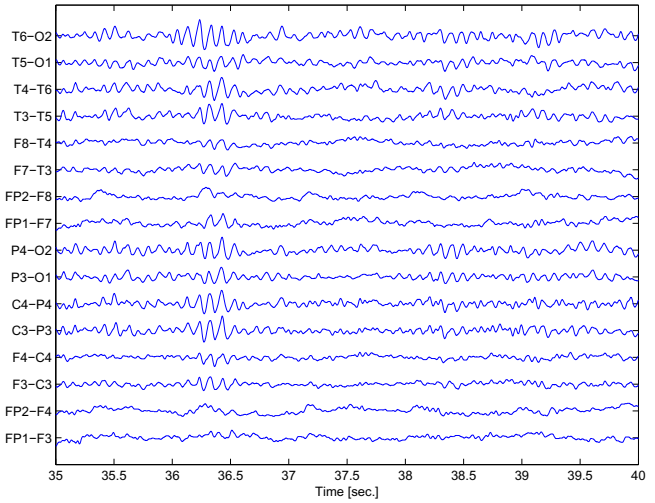


Fig. 1. Example of an EEG signal

Proper analysis and classification of EEG signals are strongly dependent on signal attributes extracted during the feature extraction process. This process can be carried out in the frequency domain. In recent years there have been elaborated methods able to present the time-frequency dependencies of a signal [11]. The most popular one is the wavelet transform and its discrete version [5,12,3]. In the 1990s there were proposed two other methods: matching pursuit [7] and the S transform [10]. The objective of this paper is to compare different methods of feature extraction in the frequency domain in the framework of EEG signals classification. The research has been conducted using real EEG recordings collected at the Ward of Neurology and Strokes of the Provincial Hospital of Zielona Góra, Poland.

2 EEG Signals

EEG signals are often used to assist the diagnosis of various neurological abnormalities. One of the frequently observed dysfunctions is epileptic seizure, which has the character of sudden discharges that cause rapid increase of the signal amplitude. The nature of such epileptic behaviour is defined as periodical spontaneous electrical discharges on the surface [14] of the cortex, which disrupt the nervous system [8]. This type of discharge may occur locally in the brain or can be observed on multiple channels simultaneously, depending on the degree of a particular dysfunction. Such electrical discharges can appear as a result of damages in the brain structure hormonal fluctuations, sudden changes in the intensity of the perception of the severity of different types of external stimuli and undesirable physiological conditions. Early diagnosis and careful analysis of EEG records may help make a proper decision concerning a suitable treatment, adjusting pharmacology and performing individual therapy.

In this study the data recorded at the Ward of Neurology and Strokes of the Provincial Hospital of Zielona Góra is analysed. These EEG records, acquired using 16 channels equipment, form a complete database of neurological disorders. The data were acquired from both epileptic patients and healthy subjects. In cooperation with the medical staff, 586 seizures from 104 patients were recorded and analysed. Simultaneously, 568 sequences from 61 healthy subjects were derived. Eventually, the database consisted of 1154 EEG sequences.

EEG signals were filtered using low-pass filter equal to 35 Hz. The sampling frequency was equal to 500 Hz (sample time equal to 0.002 sec.). An expert in clinical analysis of EEG signals (neurologist) inspected every record to score epileptic and normal sequences. Figure 1 presents an example of an EEG recording taken in the banana arrangement. In this record the expert pointed out a seizure beginning at about the 36th second.

3 Wavelet Transform

Wavelet analysis uses windows of varying size depending on the frequency. The fundamental element of the analysis is the base function called a wavelet. The wavelet is a wave-like oscillation with the amplitude starting at zero, increasing, and then decreasing back to zero. The Continuous Wavelet Transform (CWT) is defined as follows:

$$W = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \Psi \left(\frac{t-b}{a} \right) x(t) dt \tag{1}$$

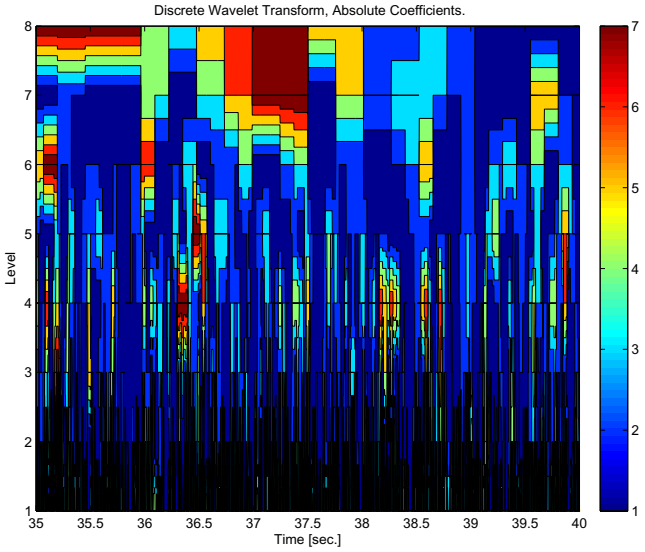


Fig. 2. EEG signal decoposition using the DWT

where Ψ is the base function, a represents the scale parameter, b is the shift parameter, and $x(t)$ is the analysed signal. The common approach to perform the Discrete Wavelet Transform (DWT) is to use the dyadic scheme to sample the base function. A particularly important property of the DWT is the multiresolution signal decomposition [6]. At each stage, the signal S is defined as a sum of approximation and detail representations. At the next stage, the approximation representation from the upper stage A_{i-1} is defined again as a sum of approximation A_i and detail representations D_i. In this way, the signal S can be represented with arbitrary accuracy using approximation and detail coefficients using a suitable number of representation stages. Figure 2 represents the wavelet decomposition of the channel T6-O2. The level represents frequency: the higher the level smaller the frequency. In this way one can analyse time-frequency dependencies in the signal. In the case considered, there are coefficients of high magnitude observed for small frequencies around 36th second. This area represents a seizure (cf. Fig. 1).

The Fourier transform cannot be used in the analysis of nonstationary signals or signals with transient characteristics. The solution is to apply Short Time Fourier Transform (STFT), but this method uses a constant window length for each frequency component. From this points of view the wavelet transform is advantageous over the classical methods of spectral analysis.

4 Matching Pursuit

Matching Pursuit (MP) is a method invented by Mallat and Zhang in 1993 [7], which involves finding the best matching projections of data onto an overcomplete dictionary D. The method represents a signal x from the Hilbert space in the form of a weighted sum of functions g_n, called atoms, taken from the dictionary D:

$$x(t) = \sum_{n=0}^{\infty} a_n g_n(t), \tag{2}$$

where a_n represents the weight for the n atom. Matching pursuit is a greedy, iteractive procedure which first finds the atom that has the biggest inner product with the signal, then substracts the contribution provided by this atom, and repeats the process until the signal is satisfactorily decomposed. The most commonly used dictionary is based on Gabor functions, which are Gaussian envelopes modulated by a harmonic signal.

In the literature one can find extensions of the basic algorithm known as multichannel matching pursuit which allow processing multidimensional data. This version of MP is an ideal choice to process EEG recordings. Figure 3 shows time-frequency representation of the channel T6-O2 processed using the basic MP, where the achieved list of coefficients representing atoms (called the book) is processed using Wigner transform. In the case considered, one can observe coefficients of high magnitude at lower frequencies. One can compare this representation with the DWT shown in Fig. 2.

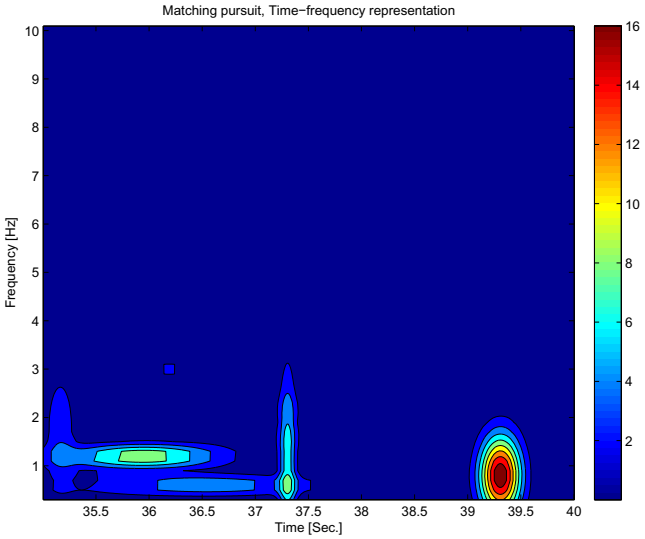


Fig. 3. Time-frequency EEG signal representation using the MP

The main problem with matching pursuit is the computational complexity. In the basic version of the method a large dictionary has to be searched at each iteraction.

5 S Transform

The S Transform (ST) is a relatively new time-frequency analysis tool developed by Stockwell in 1994 [10]. It can be viewed as a generalization of the Short-Time Fourier Transform (STFT), but instead of a window of constant size as in the STFT, it uses a scalable Gaussian window. The Stockwell transform of signal x is given as follows:

$$S(t,f) = \int_{-\infty}^{\infty} x(\tau) g(\tau - t) e^{-i2\pi f \tau} d\tau, \tag{3}$$

where $g(\tau - t)$ is the Gaussian function located at $\tau = t$ defined as

$$g(\tau - t) = |f| e^{-\pi(\tau - t)^2 f^2}. \tag{4}$$

This makes it possible to use a variable window length. As f increases, the length of the window decreases. The Stockwell transform can be viewed as a frequency dependent STFT or a phase corrected wavelet transform. Due to the fixed width of the window function, the STFT has poor time-frequency resolution. On the other hand, the wavelet transform does not retain the absolute phase information. On this basis, the Stockwell transform can be viewed as a frequency dependent STFT or a phase corrected wavelet transform.

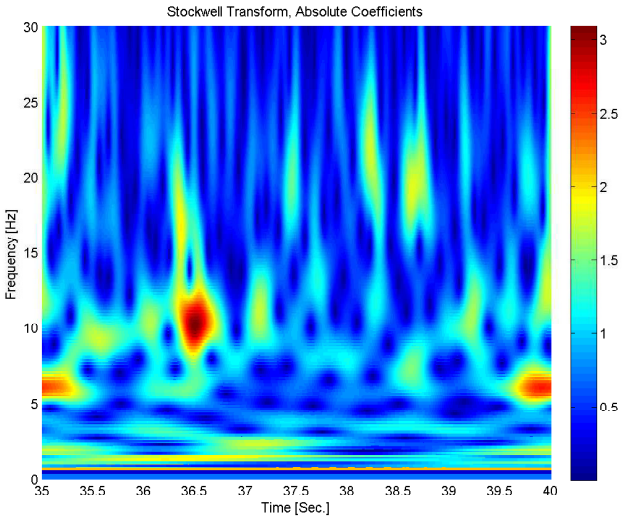


Fig. 4. Time-frequency EEG signal representation using the ST

Figure 4 represents the ST decomposition of the channel T6-O2. Contrary to the DWT the frequency is clearly represented here. In the case considered, there are coefficients of high magnitude observed for the small frequencies (about 10 Hz) around the 36th second. This area representes a seizure (cf. Figs. 1, 2, and 3).

The S transform does not have a cross-term problem and yields, a better signal clarity than the Gabor transform. The disadvantage of the S transform is complexity computation and worse clarity than, for example, that of the Wigner or Cohen distribution functions.

6 Feature Extraction

In order to perform classification, each sequence should be represented by a set of features called attributes [13]. The attributes should be properly selected to fully represent the subsequence. They make it possible to discriminate sequences one from another and to estimate the level of similarity between them. Taking into account that the EEG signal is of nonstationary character, three methods have been used to extract features: the discrete vawelet transform, matching pursuit, and the S transform.

To extract the feature, all EEG recordings representing seizures have been divided into subsequences with the length equal to 3 seconds. Taking into account that the sampling time is equal to 0.002 seconds, this leads to the number of samples equal to 1500.

6.1 Wavelet Tranform

Each channel of the EEG recording sequence was represented with the 7-th level of multi-resolution representation using the Daubechies 3 (db3) wavelet. In this way each level represents a suitable range of frequencies. After the decomposition one obtains several subsequences of different length depending on the scale. Increasing the scale the sequence length is decrasing. The specification of the achieved transformation components is presented in Table 1. The DWT results in

Table 1. Specification of the DWT

component	frequency range [Hz]	sequence length [samples]
detail 1	$(250 - 500)$	753
detail 2	$(125 - 250)$	379
detail 3	$(62.5 - 125)$	192
detail 4	$(31.25 - 62.5)$	98
detail 5	$(15.63 - 31.25)$	51
detail 6	$(7.8 - 15.63)$	28
detail 7	$(3,9 - 7.81)$	16
approximation 7	$(0 - 3.9)$	16

a quite large number of samples per each component. Therefore, each component is further processed in order to reduce the number of attributes. In this work the following methods have been used:

- deriving statistical parameters of each component in the form of the mean value, standard deviation, median and mode,
- deriving component energy,
- deriving the entropy of each component.

Each method gives a different number of attributes. In the case of statistical parameters, we have 512 attributes (16 channels $\times$ 8 components $\times$ 4 indicators), and in the case of signal energy or signal entropy, we have 128 attributes (16 channels $\times$ 8 components $\times$ 1 indicator).

6.2 Matching Pursuit

In the paper, the socalled multichannel matching pursuit has been applied. This version of MP allows approximating all channels of the EEG recording simultaneously using the same book. In the present study, the Gabor dictionary was applied when the number of atoms was set to 10. In order to select the attributes we have used:

- parameters of the decomposition stored in the book,
- statistical parameters of the each atom in the form of the mean value, standard deviation, median and mode.

In the case of rare data stored in the book, we have 190 attributes (10 atoms × 19 parameters), and in the case of the statistical parameters, we have 40 attributes (10 atoms × 4 indicators).

6.3 S Transform

By analysing the EEG recordings we have decided that the S transform should be performed for the frequency range 0-100 Hz. Moreover, the frequency sampling rate was equal to 10. Using such settings, each channel is decomposed into 11 components, each component has the full length of 1500 samples. Threrefore, similarily like in the case of the DWT each component has been processed to obtain attributes

- calculating statistical parameters of each component in the form of the mean value, standard deviation, median and mode,
- calculating component energy,
- calculating the entropy of each component.

In the case of statistical parameters, we have 704 attributes (16 channels × 11 components × 4 indicators), and in the case of the signal energy or signal entropy, we have 176 attributes (16 channels × 11 components × 1 indicator).

7 Classification

The K Nearest Neighbours (KNN) algorithm is a method for classifying patterns based on the closest training examples in the feature space [1]. This algorithm is the simplest method out of all machine learning algorithms. The pattern is classified by a majority vote of its neighbours, with the pattern being assigned to the class most common amongst its K nearest neighbours. In our previous work [9], it was shown that the KNN classifier performed the best out of many different classifiers tested during EEG signal classification. That is the reason for using KNN in this work. The settings are as follows: $k = 1$, and the Euclidean distance is used as the distance metric.

The classification process using each classifier has been repeated 1000 times, calculating the best classification rate Q_{min}, the worst one Q_{max}, and the average Q_{av} (mean value over 1000 tests). At each run the entire set of samples has been divided into the training set containing 200, 350 and 500 samples and the testing one containing the rest of patterns. The classification results are shown in Tables 2 - 4, where the best rates are marked in bold face.The abbreviations used in tables are as follows:

- MP_1 : attributes are rare coefficients of the book (see Section 6.2),
- MP_2 : attributes are calculated as statistical parameters (see Section 6.2),
- DWT_1 : attributes are calculated as statistical parameters (see Section 6.1),
- DWT_2 : attributes are derived using components energy (see Section 6.1),
- DWT_3 : attributes are derived using components entropy (see Section 6.1),

Table 2. Classification results, size of the training set = 200

training set size 200	MP_1	MP_2	ST_1	ST_2	ST_3	DWT_1	DWT_2	DWT_3
Q_{av}	61	81	87.7	87.9	86.6	79.8	**88.2**	87.2
Q_{max}	66	86.8	93.1	93	92.4	85	**93.2**	92.2
Q_{min}	55	74.7	80.8	80.7	80.4	73.2	**82.6**	78.5

Table 3. Classification results, size of the training set = 350

training set size 350	MP_1	MP_2	ST_1	ST_2	ST_3	DWT_1	DWT_2	DWT_3
Q_{av}	64	85.5	93.6	93.7	92.8	85.3	93.8	**94.1**
Q_{max}	69.1	90	96.4	97.6	96.5	89,8	97.3	**98.6**
Q_{min}	58.9	80.4	86.7	87.9	88.7	79.6	**89**	88.6

Table 4. Classification results, size of the training set = 500

training set size 500	MP_1	MP_2	ST_1	ST_2	ST_3	DWT_1	DWT_2	DWT_3
Q_{av}	66	88.4	96	96.1	95.6	88.7	96.3	**96.9**
Q_{max}	70.6	92.7	98.3	98.3	98.5	92.1	98.6	**99.2**
Q_{min}	60.8	83.9	91.3	92.4	91.4	82.4	92	**92.7**

- ST_1 : attributes are calculated as statistical parameters (see Section 6.3),
- ST_2 : attributes are derived using components energy (see Section 6.3),
- ST_3 : attributes are derived using components entropy (see Section 6.3).

In each case considered the classification system based on attributes selected using the discrete wavelet transform gives the best results. However, for the smaller training sets a slightly better performation is observed in the case of a classifier based on attributes derived using the energy of each component. For the other cases, classifiers working with attributes derived using the entropy of the components give better results.

The worst performance is observed for MP methods, but it should be kept in mind that, in this case, attributes have been selected differently than in the case of the DWT or the ST. To definitely judge MP, further research should be carried out focusing on other ways of feature acquiring.

Concerning the ST, the achieved results are slighty worse than these obtained by the DWT but differences are insignificant. In general, the main problem with the ST is how to properly select the sampling frequency rate. This choice is only a compromise between the quality of time-frequency representation and the complexity of data representation.

8 Conclusions

The paper presents a comparison of methods such as the discrete wavelet transform, the matching pursuit algorithm and the S transform for feature extraction

and classification of neurological disorders. The results are very promising and confirm the validity of the experiments. Improving the existing diagnostic tools using the latest technology aims to make accurate assessment of the clinical diagnoses posed. Close cooperation with specialists in the field of electroencephalography the Ward of Neurology and Stroke in Zielona Góra will allow the implementation of further research in the field of biomedical signal analysis. The created tools based on advanced algorithms and technologies available to support clinical diagnosis can often have allow avoiding mistakes in diagnostic assessment.

References

1. Duda, R.O., Hart, P.E., Stork, D.H.: Pattern Classification, 2nd edn. Wiley Interscience (2000)
2. Faul, S., Boylan, G., Connolly, S.: An evaluation of automated neonatal seizure detection methods. Clin. Neurophysiol. 116, 1533–1541 (2005)
3. Guo, L., Rivero, D., Pazos, A.: Epileptic seizure detection using multiwavelet transform based approximate entropy and artificial neural networks. J. Neurosci. Methods 193, 156–163 (2010)
4. James, C.J., Jones, R.D., Bones, P.J., Carroll, G.J.: Detection of epileptiform discharges in the eeg by a hybrid system comprising mimetic, self-organized artificial neural network, and fuzzy logic stages. Clin. Neurophysiol. 12, 2049–2063 (1999)
5. Kahn, Y., Gotman, J.: Wavelet based automatic seizure detection in intracerebral electroencephalogram. Clin. Neurophysiol. 114, 898–908 (2003)
6. Mallat, S.: A theory for multiresolution signal decomposition: The wavelet representation. IEEE Trans. Pattern Anal. Mach. Intell. 11(7), 674–693 (1989)
7. Mallat, S., Zhang, Z.: Matching pursuits with time-frequency dictionaries. IEEE Trans. Signal Processing 41(12), 3397–3415 (1993)
8. McSharry, P., He, T., Smith, L.: Linear and non-linear methods for automatic seizure detection in scalp electro-encephalogram recordings. Med. Biol. Eng. Comput. 40, 447–461 (2002)
9. Patan, K., Rutkowski, G.: Analysis and classification of eeg data: An evaluation of methods. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 310–317. Springer, Heidelberg (2012)
10. Stockwell, R.G., Mansinha, L., Lowe, R.P.: Localization of the complex spectrum: The s transform. IEEE Trans. Signal Processing 44(4), 998–1001 (1996)
11. Świercz, E.: Classification in the gabor time-frequency domain of non-stationary signals embedded in heavy noise with unknown statistical distribution. J. Applied Mathematics and Computer Science 20(1), 135–147 (2010)
12. Übeyli, E.D.: Combined neural network model employing wavelet coefficients for eeg signals classification. Digital Signal Processing 19, 297–308 (2009)
13. Woźniak, M., Krawczyk, B.: Combined classifier based on feature space partitioning. J. Applied Mathematics and Computer Science 22(4), 855–866 (2012)
14. Yeung, N., Bogacz, R., Holroyd, C.B., Cohen, J.D.: Detection of synchronized oscillations in the electroencephalogram: An evaluation of methods. Psychophysiology 41, 822–832 (2004)

GP-Pi: Using Genetic Programming with Penalization and Initialization on Genome-Wide Association Study

Ho-Yin Sze-To[1,⋆], Kwan-Yeung Lee[1], Kai-Yuen Tso[1], Man-Hon Wong[1], Kin-Hong Lee[1], Nelson L.S. Tang[2,⋆], and Kwong-Sak Leung[1,⋆]

[1] Department of Computer Science and Engineering
{hyszeto,kylee0,kytsomhwong,khlee,ksleung}@cse.cuhk.edu.hk
http://www.cse.cuhk.edu.hk
[2] Laboratory for Genetics Disease Susceptibility,
Li Ka Shing Institute of Health Sciences
nelsontang@cuhk.edu.hk
http://www.lihs.cuhk.edu.hk
The Chinese University of Hong Kong,
Shatin, N. T., Hong Kong, China

Abstract. The advancement of chip-based technology has enabled the measurement of millions of DNA sequence variations across the human genome. Experiments revealed that high-order, but not individual, interactions of single nucleotide polymorphisms (SNPs) are responsible for complex diseases such as cancer. The challenge of genome-wide association studies (GWASs) is to sift through high-dimensional datasets to find out particular combinations of SNPs that are predictive of these diseases. Genetic Programming (GP) has been widely applied in GWASs. It serves two purposes: attribute selection and/or discriminative modeling. One advantage of discriminative modeling over attribute selection lies in interpretability. However, existing discriminative modeling algorithms do not scale up well with the increase in the SNP dimension. Here, we have developed GP-Pi. We have introduced a penalizing term in the fitness function to penalize trees with common SNPs and an initializer which utilizes expert knowledge to seed the population with good attributes. Experimental results on simulated data suggested that GP-Pi outperforms GPAS with statistically significance. GP-Pi was further evaluated on a real GWAS dataset of Rheumatoid Arthritis, obtained from the North American Rheumatoid Arthritis Consortium. Our results, with potential new discoveries, are found to be consistent with literature.

Keywords: Genome-Wide Association Study, Genetic Programming, Penalization, Initialization, Rheumatoid Arthritis.

1 Introduction

The advancement of chip-based technology has enabled the measurement of millions of DNA sequence variations across the human genome [1, 2]. The by far

⋆ To whom correspondence should be addressed.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 330–341, 2013.

most common type of such genetic variations are single nucleotide polymorphisms (SNPs), which occur when a different base alternative exists at a single base pair position [3]. In this paper, we focus on studying SNPs, which are categorical variables with 3 outcomes. It is anticipated that at least one SNP occurs approximately every 100 nucleotides across the 3×10^9 nucleotides in human genome [4]. Genome-Wide Association Study (GWAS) is to find out which of the many differences in the genotypes are associated with the phenotypes, or, more specifically, to determine which of the many SNPs are useful for predicting the risk for common diseases, particularly genetic diseases [5]. The role of Computer Science and Bioinformatics is to develop efficient and effective algorithms to identify the disease-associated SNPs. It is a challenging task due to the non-linear mapping between genotypes and phenotypes. SNPs need to be considered jointly in learning algorithms rather than individually. This non-linearity is what we call Epistasis. Here, intelligent algorithms are needed to solve the problem.

1.1 Concept Difficulty

It is believed that high-order interactions of SNPs, not individual SNP, are culprits of complex diseases such as cancer. The challenge of GWAS is to sift through high-dimensional datasets to find out particular combinations of SNPs that are predictive of diseases. Researchers call this a needle-in-a-haystack problem. That is, there may be a particular combination of SNPs which fits well together with a non-linear function and yields good performance when they are used as predictors, e.g. SNP_1 AND SNP_2. However, when these SNPs are considered individually, they may not look different from irrelevant SNPs. Here, the learning algorithm is searching for a genetic haystack, i.e. the number of candidate SNP interactions is tremendous. For J SNPs and pair-wise interaction, J^2 pairs are needed to be considered for epistasis, which is referred to the effect of one locus depending on the genotype of another locus. In general, for J SNPs and K-way interactions, there are O (J^K) candidates [6]. It is computationally infeasible to consider all possible groups of SNP interactions.

1.2 Genetic Programming

Genetic programming (GP) is an automated discovery tool based on Darwinian evolution and natural selection. The goal of GP is to evolve the fittest computer programs to solve problems. Starting from a population of randomly generated computer programs, the GP algorithm evaluates all programs. The good programs are selected, recombined and mutated to form new programs. The process will be terminated after a number of generations or the target fitness is achieved. Genetic programming and its many variations have been applied successfully to a wide range of domains including but not limit to robot vision [7], computational finance [8], drug discovery [9] and motif discovery [10, 11].

1.3 Genetic Programming in GWAS

GP has also been widely applied in GWAS. It serves two purposes: attribute selection [4, 12] and discriminative modeling. For attribute selection, studies have demonstrated that GP is a successful wrapper approach in selecting a few important SNPs among thousands of irrelevant ones. It is also found that the use of expert knowledge can significantly improve the performance of GP algorithms [4, 12, 13].

While the above methods demonstrate satisfactory result in attribute selection, they lack the ability to learn a classification model directly, i.e. discriminative modeling. Discriminative modeling is to learn a classification model to best predict samples susceptibility to disease. While attribute selection may also be performed, the learned model can be directly applied on unseen data to perform classification. One advantage of discriminative modeling approach is that, given the classification model is interpretable, biologists can judge whether the algorithmic output is consistent to biological knowledge and whether it has real-world application value.

GPDTI (Genetic Programming Decision Tree Induction) [14] and GPAS (Genetic Programming for Association Studies) [3] are two discriminative algorithms. They both adopt decision-tree like models to represent their solutions. GPDTI uses basic expression trees and operators to model the problem. GPAS attempts to detect DNFs associated with the response directly by employing multi-valued logic. However, neither GPDTI nor GPAS utilize expert knowledge in guiding GP. As suggested by [4], GP performs no better than random search when expert knowledge is not provided. Thus, they cannot be scaled well with the increase in SNP size.

1.4 Paper Layout

This paper is organized as follows: the proposed methods are detailed in Section 2; data simulation and analysis are reported in Section 3; experimental results on simulated data are investigated in Section 4; experimental results on a real GWAS dataset of Rheumatoid Arthritis are illustrated in Section 5; the whole article is discussed and concluded in Section 6.

2 A Novel Discriminative Model: GP-Pi

In this research, we would like to design a new GP algorithm for discriminative modeling by utilizing expert knowledge. We have specifically developed an attribute weighting initializer which makes use of ReliefF [15] as expert knowledge. Although similar work [12] has been used in attribute selection approach (which used Tuned ReliefF), research has yet to prove if it works in discriminative modeling approaches. To enhance the interpretability of our model, we chose not to follow [12] to adopt MDR [16], an exhaustive feature construction algorithm, as one of the function sets. To prevent premature convergence to local optima, we

maintain the diversity of population by penalizing similar individuals in their fitness. The introduction of a penalizing term in the fitness function is a novel GP application in GWAS. We evaluate our algorithm, GP-Pi (P stands for penalization and I stands for initialization), by comparing it against GPAS, which is publicly available at `http://ls2-www.cs.uni-dortmund.de/~nunkesser/`. Experimental results suggest that GP-Pi outperforms GPAS with statistically significance.

2.1 Genetic Programming Methods

We have adopted Genetic Programming to evolve models to model SNP-SNP interactions. We have used a crossover probability of 0.9, a mutation probability of 0.05, and a no-operation probability of 0.05. The maximum depth of a tree is 5. We have two function sets (And, Or) and two terminal sets (Equal, Not Equal). The overall GP parameters are summarized in Table 1. Comparing to the previous work, we contributed in two aspects. First, we introduced a penalizing term in the fitness function to penalize tree with common SNPs. Second, we developed an initializer which utilizes expert knowledge to initialize a good population. GP-Pi has been implemented in ECJ [17].

Tree Representation of Solutions. We have followed the approach suggested by [3] to express SNP-SNP interactions in logic expressions. However, we do not force the logic expression to be disjunctive normal form (DNF). Figure 1 depicts how we model GP expression trees to SNP-SNP interactions.

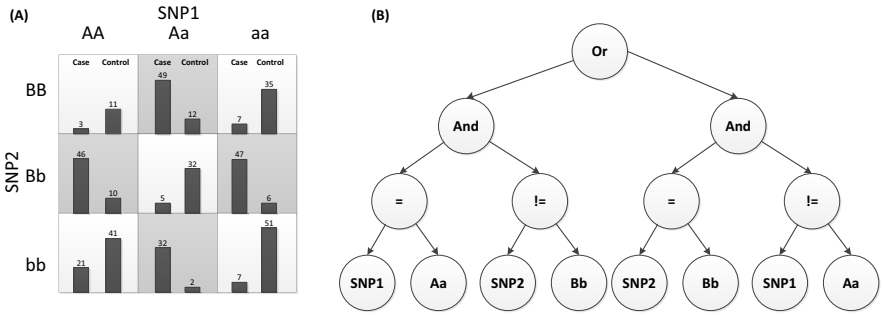


Fig. 1. (A) illustrates a SNP-SNP interaction in a GWAS. The left bars within each cell represent the number of cases (with diseases) while the right bars represent the number of controls (without diseases). Dark-shaded cells are high risk while the light-shaded cells are low risk. (B) is a GP tree modeling the predictive logic of the dark-shaded cells (high-risk combinations of SNP patterns). Hence, GP algorithm can be designed to evolve expression trees like (B) to model SNP-SNP interactions like (A).

Table 1. Summary of GP Parameters

Item	Parameter
Population Size	4096
Generations	40
Crossover	Single-point subtree
Crossover frequency	0.9
Mutation frequency	0.05
Fitness Function	$f_i = E_i + \frac{N_i}{\alpha} + P_i$
Selection	Sevens tournament
Function Set	And, Or
Terminal Set	$=, \neq$
Maximum Tree Depth	5

Initialization. We have developed a probabilistic initializer which utilizes expert knowledge to select attributes (SNPs). Here, we describe the mechanism of the initialization. First, we compute a score S for each attribute m in tournament. The attribute m with the highest score is selected. The equation is shown below:

$$S_m = R_m + U_m + \beta \times K \tag{1}$$

where R_m is the ReliefF score (See section 3), U_m is the usage score, β is a constant determined in runtime. If a random number V is larger than a threshold t_{relief} (0.8 is default), β will be 6, otherwise 1. K is a random number which is randomized in each tournament. Here, all random numbers range from 0 to 1 inclusively. The usage score is to keep the population as diverse as possible. The more an attribute appearing in the population, the lower its score is. The equation of Usage score is shown below:

$$U_m = \frac{L - u_m}{L} \tag{2}$$

where L is the highest number of appearance among all attributes and u_m is the number of appearance of m.

Penalization. Referencing a Koza style [18] of fitness function defined in [14], we introduced a penalization term to penalize trees with a certain degree of common SNPs and with similar number of nodes. The fitness function is shown below:

$$f_i = E_i + \frac{N_i}{\alpha} + P_i \tag{3}$$

where i is an index to an individual, E_i is the classification error, N_i is the number of nodes, α is the parsimony constant [19] (2 is default) and P_i is the penalization term. P_i is set to 100 if the following criterion is satisfied:

$$\frac{|N_i - N_j|}{m} < t_1 \wedge \frac{c}{m} > t_2 \tag{4}$$

where t_1 is a threshold (0.1 is default), t_2 is another threshold (0.9 is default), j is any individual within the population except i, c is the number of SNPs that are included by the tree expressions of both i and j, m is the average number of nodes between i and j. if the above criterion is not satisfied, P_i is set to 0.

In other words, every individual is compared against each other in every generation. Similar individuals are penalized based on the above procedure. This process helps maintain a wide diversity of population and prevents against premature convergence.

2.2 Expert Knowledge Guiding the Search

The use of expert knowledge is to provide an external measure of attribute quality to guide our search to overcome the needle-in-a-haystack problem [4]. Relief [15] is one feature selection algorithm which has the capability. The basic idea of Relief is to iteratively estimate feature weights according to their ability to discriminate between neighboring patterns. ReliefF [20] is an improvement in robustness of relief by considering the nearest k neighbors but not only the nearest one. Both Relief and ReliefF can capture attribute interactions because they use the entire vector of values to find nearest neighbor(s). However, they are also susceptible to noise attributes. Tuned ReliefF [21] is an improvement in susceptibility to noise attributes of ReliefF by systematically remove the low quality attributes so that the ReliefF score of the remaining attributes can be re-estimated. Throughout this research, we adopted ReliefF but not Tuned ReliefF as our expert knowledge provider, for similar performance but less computational time. This is different from the similar work [12] which adopted Tuned ReliefF.

3 Data Simulation and Analysis

A simulation study was performed to evaluate our GP algorithm in Genome-Wide Association Study (GWAS). Using GAMETES [22], a GWAS dataset generator, we developed 4 penetrance functions (i.e. genetic model) which define a probabilistic relationship between genotype and phenotype (or the susceptibility to disease that depends only on genotype but not any other effects). Each model has 2 functional SNPs with minor allele frequency of 0.2. Also, each model is corresponded to a heritability (the effect size of functional SNPs) of 0.1, 0.2, 0.3, 0.4 respectively. An example is shown in Table 2. Based on the above model, each pair of functional SNPs was then combined within a genome-wide set of 98, 998 randomly generated SNPs to form a total of 100, 1000 attributes. Keeping a balanced ratio of cases and controls among 2000 samples, we generated 10 replicates for each parameter setting. A total of 80 datasets were generated and analyzed. All datesets with full precision are available upon request.

For each dataset, we ran our GP algorithm independently 10 times. For each parameter setting, we had in total 100 runs (10 replicates $\times$ 10 runs). For each parameter setting, we counted the number of times that the correct two functional SNPs were selected as nodes in the best GP tree model. This count was

Table 2. An Example Epistasis Model with Heritability 0.3

	AA (0.64)	Aa (0.32)	aa (0.04)
BB (0.64)	0.515	0.913	0.779
Bb (0.32)	0.934	0.124	0.383
bb (0.04)	0.614	0.712	0.792

expressed as a percentage, which was an estimation of the power of the method. Based on this count, we can estimate how often our GP algorithm can get the right answer if there is. We compared our result against GPAS (Genetic Programming for Association Study), a GP algorithm similar to ours but neither does it exploit expert knowledge nor penalize individuals. For each dataset, we ran GPAS on all simulated datasets with SNP size of 100 and 1000 for 10 independent runs, in total 100 runs for each parameter setting. 100,000 generations were allowed on each run. We consider the output of each run of GPAS as correct if the best 5 individuals contain the two functional SNPs. The power, i.e. the percentage that the algorithm outputs correct result, can then be estimated for GPAS. We compared the power of our GP algorithm against GPAS on estimation of power using chi-square test of independence. Results are considered statistically significant when the p-value of the chi-square test statistic is ≤ 0.05.

4 Experimental Results on Simulated Data

The power (the percentage that the algorithm identifies the correct two functional SNPs) for each method across heritability of 0.1, 0.2, 0.3 and 0.4 with a SNP size of 100 and 1000 is summarized in Figure 2. Each bar on the plots represents the power over the 100 runs of each parameter setting, or, in other words, represents the percentage that the algorithm can select the two functional SNPs as nodes of trees. Our algorithm was compared against GPAS in these experiments.

While our algorithm had a robust performance on 100 SNP size across different heritiabilities, it also had a satisfactory performance on 1000 SNP size. On 100 SNP size, we nearly achieved 100% power. On 1000 SNP size, we still had a certain percentage of power even when the heritability dropped to 0.1. In terms of comparison, our algorithm outperformed GPAS in terms of power on both SNP size of 100 (P = 5.16E-10 <0.05) and SNP size of 1000 (P = 4.85E-36 <0.05). The size of the search space is approximately 5000 (100 SNPs choose 2) and 500,000 (1000 SNPs choose 2). With a population size of 4096 and 40 generations, GP-Pi is exploring at most 3300% (SNP Size: 100) and 33% (SNP Size: 1000) of the search space. It should be noted that GPAS is allowed to have 100,000 generations on each run. Assume it has a population size of 10, GPAS is exploring at most 20000% (SNP: 100) and 200% (SNP: 1000) of the search space. While the best 5 individuals on each run of GPAS are extracted to examine its correctness, only the best individual is outputted to be examined in GP-Pi.

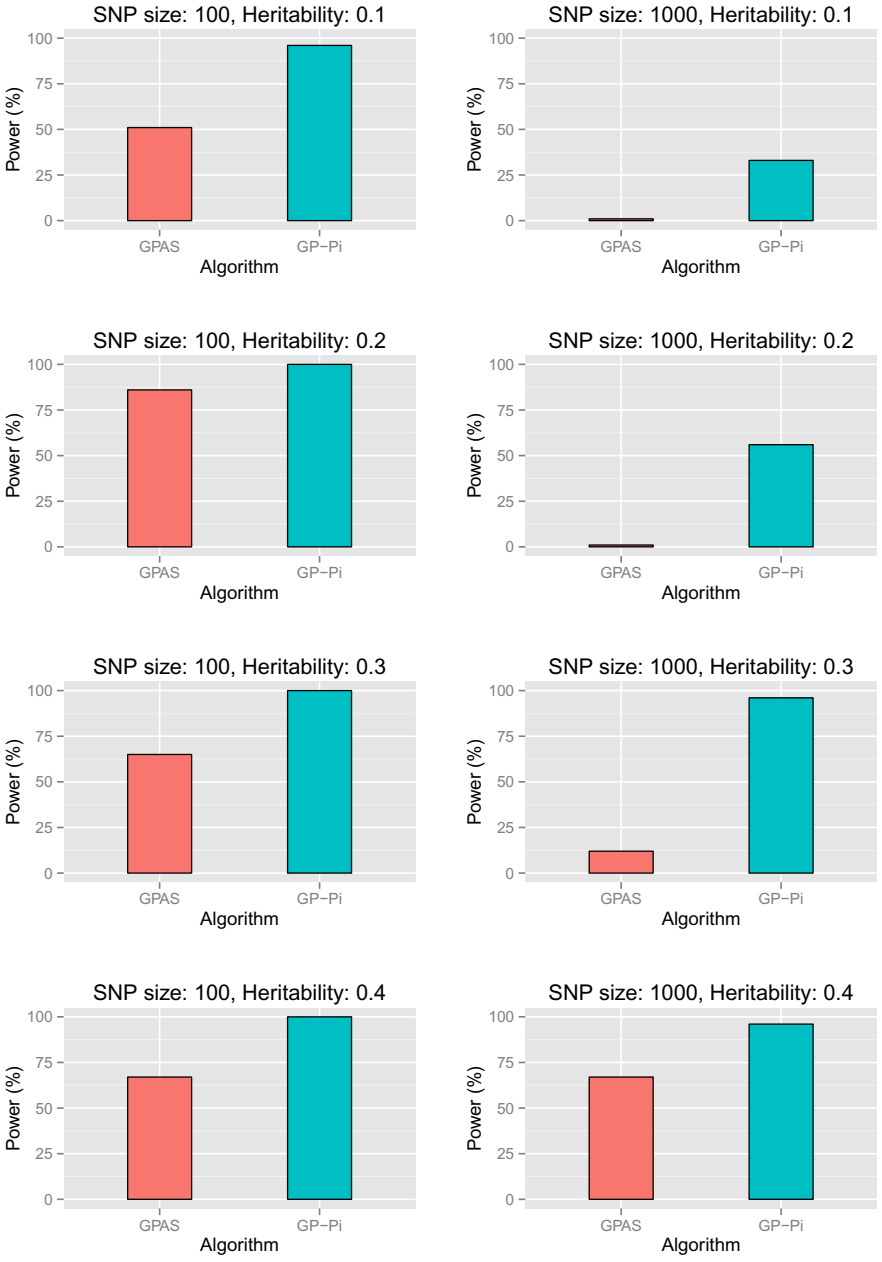


Fig. 2. The power (the percentage that the algorithm identifies the correct two functional SNPs) of GPAS and GP-Pi across heritability of 0.1, 0.2, 0.3 and 0.4 with a SNP size of 100 and 1000 is summarized above. GP-Pi outperformed GPAS in terms of power on both SNP size of 100 (P = 5.16E-10 <0.05) and SNP size of 1000 (P = 4.85E-36 <0.05).

5 Experimental Results on Real Data

In this section, we demonstrate the capability of GP-Pi on handling with real data. We applied GP-Pi on a real GWAS dataset. Our GWAS dataset is obtained from the North American Rheumatoid Arthritis Consortium (`http://www.naracdata.org/`). There are 2062 patients in total, 868 of which are cases which suffer from Rheumatoid Arthritisand the remaining (1194) are controls. There are in total 545080 SNPs in this study, which constitutes a gigantic search space.

5.1 Data Preprocessing

There are three types of genotypes: 'AA', 'Aa' and 'aa'. They were encoded as 1, 2, 3 respectively. SNPs with more than 15% of missing values were filtered. The missing values of the remaining SNPs were chosen randomly. As the number of SNPs in our search space is tremendous, we select the top 1000 SNPs with top ReliefF scores.

5.2 Data Mining and Parameter Setting

The dataset was shuffled and split into testing data and training data on a ratio of 1 to 9. GP-Pi was run on training data under a 10-fold cross-validation to learn classification models. In total, 10 classification models were learned and evaluated on testing data. It should be noted that the same set of GP parameters, summarized in Table 1 were used in these experiments.

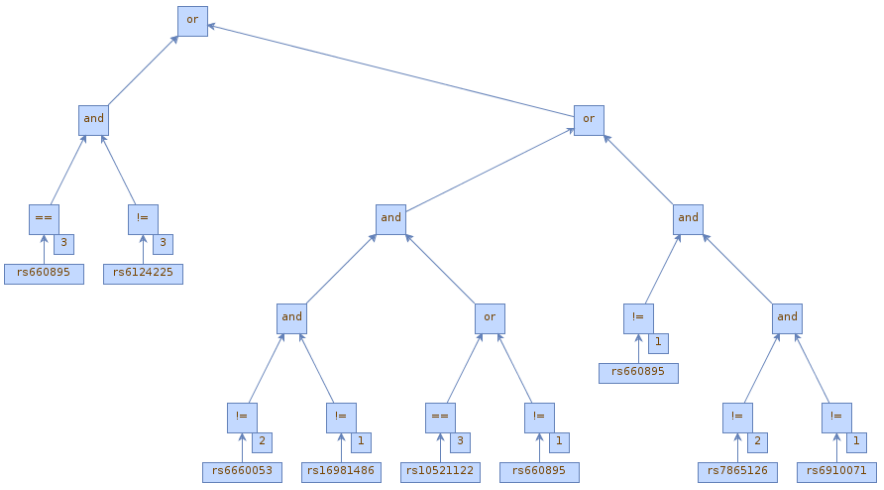


Fig. 3. The above is the best discriminative model outputted by GP-Pi on a real GWAS dataset of Rheumatoid Arthritis to classify if a patient is a case or a control. It has achieved a sensitivity of 0.741 and a specificity of 0.806. Both rs660895 and rs6910071 are confirmed to be risk loci with literature support respectively [23, 24]. rs7865126, with its susceptibility to tuberculosis infection [25, 26], may be a new discovery.

5.3 Result and Analysis

Among the 10 models, the best one is selected based on its performance in validation data . It is then evaluated on testing data and the result is as follows: Accuracy: 0.790, Precision: 0.741, Recall: 0.769, F-measure: 0.754, Specificity: 0.806. The best model is demonstrated in Figure 3. It should be noted that both rs660895 and rs6910071 are confirmed to be rheumatoid arthritis risk loci with literature support respectively [23, 24]. In addition, rs7865126 may be a novel discovery of rheumatoid arthritis risk loci.

6 Discussion and Conclusion

In this paper, our objective is to show discriminative modeling on GWAS with GP can be improved by utilizing expert knowledge. We have developed an initializer which exploits ReliefF score as expert knowledge to seed the initial population with good attributes, and introduced a penalization term into the fitness function to penalize trees with too many common SNPs. Experiments have been run on simulated data and compared against our performance with GPAS. Results on simulated data has shown that GP-Pi outperformed GPAS with statistical significance, where GPAS is a discriminative modeling GP with neither penalization nor initialization. It shows that our method plays an important role in improving performance on guiding the search.

We also have applied GP-Pi on a real GWAS dataset of Rheumatoid Arthritis to prove the applicability of GP-Pi. Our method performed up to expectation in that (1) it picked up both HLA SNPs which are known to have the largest effect on RA susceptibility (rs660895 and rs6910071), (2) it demonstrated the ability of the method to delineate both local and inter-chromosomal interaction effects. For local interaction, it represents the effect of haplotype and is exemplified by the rightmost branch (involving rs660895, and rs6910071). In addition, this branch also involves an interaction with rs7865126 which is located in another chromosome. The leftmost branch is another example of inter-chromosomal (inter-genic) interaction which involves rs660895 (a HLA gene) and rs6124225. The decision tree also showed internal consistence, for example, in all branches where rs660895 was involved, either "3" or !="1" genotypes were found to associated with disease. In addition, rs7865126 was found to be a new interaction partner with SNPs in HLA loci. rs7865126 is located within the gene coding for a subunit of the augmin complex. Recently, Png et al [25] and Li et al [26] reported SNPs in this gene are related to susceptibility to tuberculosis infection and possibly to innate response.

In this paper, we have shown that discriminative modeling on GWAS with GP can be significantly improved by penalization and initialization . We have also illustrated that GP performs not only attribute selection but also discriminative modeling well. Our next question is about the limits of these algorithms. Can these algorithms process millions of genetic variations directly without any kind of filtering? If not, what are the alternatives? These questions may be challenging to answer. However, we believe that with the exploding data volume of

human genetics, intelligent algorithms like GP will be in critical need and keep on playing an important role in the future development of GWAS.

References

1. Hirschhorn, J., Daly, M.: Genome-wide association studies for common diseases and complex traits. Nature Reviews Genetics 6(2), 95–108 (2005)
2. Wang, W., Barratt, B., Clayton, D., Todd, J.: Genome-wide association studies: theoretical and practical concerns. Nature Reviews Genetics 6(2), 109–118 (2005)
3. Nunkesser, R., Bernholt, T., Schwender, H., Ickstadt, K., Wegener, I.: Detecting high-order interactions of single nucleotide polymorphisms using genetic programming. Bioinformatics 23(24), 3280–3288 (2007)
4. Moore, J., White, B.: Genome-wide genetic analysis using genetic programming: The critical need for expert knowledge. In: Genetic Programming Theory and Practice IV, pp. 11–28 (2007)
5. Reich, D., Lander, E.: On the allelic spectrum of human disease. TRENDS in Genetics 17(9), 502–510 (2001)
6. Moore, J., Asselbergs, F., Williams, S.: Bioinformatics challenges for genome-wide association studies. Bioinformatics 26(4), 445–455 (2010)
7. Martin, M.C.: Genetic programming for real world robot vision. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 67–72. IEEE (2002)
8. Chen, S.H.: Genetic algorithms and genetic programming in computational finance. Springer (2002)
9. Langdon, W., Barrett, S.: Genetic programming in data mining for drug discovery. In: Evolutionary Computation in Data Mining, pp. 211–235 (2005)
10. Lo, L., Chan, T., Lee, K., Leung, K.: Challenges rising from learning motif evaluation functions using genetic programming. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, pp. 171–178. ACM (2010)
11. Wong, K., Peng, C., Wong, M., Leung, K.: Generalizing and learning protein-dna binding sequence representations by an evolutionary algorithm. Soft Computing-A Fusion of Foundations, Methodologies and Applications 15(8), 1631–1642 (2011)
12. Greene, C., White, B., Moore, J.: Sensible initialization using expert knowledge for genome-wide analysis of epistasis using genetic programming. In: IEEE Congress on Evolutionary Computation, CEC 2009, pp. 1289–1296 (2009)
13. Greene, C., White, B., Moore, J.H.: An expert knowledge-guided mutation operator for genome-wide genetic analysis using genetic programming. In: Rajapakse, J.C., Schmidt, B., Volkert, L.G. (eds.) PRIB 2007. LNCS (LNBI), vol. 4774, pp. 30–40. Springer, Heidelberg (2007)
14. Estrada-Gil, J., Fernández-López, J., Hernández-Lemus, E., Silva-Zolezzi, I., Hidalgo-Miranda, A., Jiménez-Sánchez, G., Vallejo-Clemente, E.: Gpdti: A genetic programming decision tree induction method to find epistatic effects in common complex diseases. Bioinformatics 23(13), i167–i174 (2007)
15. Kira, K., Rendell, L.: A practical approach to feature selection. In: Proceedings of the Ninth International Workshop on Machine Learning, pp. 249–256. Morgan Kaufmann Publishers Inc. (1992)
16. Hahn, L., Ritchie, M., Moore, J.: Multifactor dimensionality reduction software for detecting gene–gene and gene–environment interactions. Bioinformatics 19(3), 376–382 (2003)

17. Luke, S., Panait, L., Balan, G., Paus, S., Skolicki, Z., Bassett, J., Hubley, R., Chircop, A.: Ecj: A java-based evolutionary computation research system (2007)
18. Koza, J., James, P.: Rice, genetic programming (videotape): the movie (1992)
19. Bleuler, S., Brack, M., Thiele, L., Zitzler, E.: Multiobjective genetic programming: Reducing bloat using spea2. In: Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, pp. 536–543. IEEE (2001)
20. Wiskott, L., Fellous, J., Kruger, N., Malsburg, C.: Estimating attributes: analysis and extension of relief. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
21. Moore, J.H., White, B.C.: Tuning relieff for genome-wide genetic analysis. In: Marchiori, E., Moore, J.H., Rajapakse, J.C. (eds.) EvoBIO 2007. LNCS, vol. 4447, pp. 166–175. Springer, Heidelberg (2007)
22. Urbanowicz, R., Kiralis, J., Sinnott-Armstrong, N., Heberling, T., Fisher, J., Moore, J.: Gametes: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures. BioData mining 5(1), 16 (2012)
23. Gorman, J., David-Vaudey, E., Pai, M., Lum, R., Criswell, L.: Particular hla–drb1 shared epitope genotypes are strongly associated with rheumatoid vasculitis. Arthritis & Rheumatism 50(11), 3476–3484 (2004)
24. Stahl, E.A., et al.: Genome-wide association study meta-analysis identifies seven new rheumatoid arthritis risk loci. Nat Genet. 42(6), 508–514 (2010)
25. Png, E., Alisjahbana, B., Sahiratmadja, E., Marzuki, S., Nelwan, R., Balabanova, Y., Nikolayevskyy, V., Drobniewski, F., Nejentsev, S., Adnan, I., et al.: A genome wide association study of pulmonary tuberculosis susceptibility in indonesians. BMC Medical Genetics 13(1), 5 (2012)
26. Li, S., Wang, L., Berman, M., Kong, Y.Y., Dorf, M.E.: Mapping a dynamic innate immunity protein interaction network regulating type i interferon production. Immunity 35(3), 426–440 (2011)

New Approach for the On-Line Signature Verification Based on Method of Horizontal Partitioning

Marcin Zalasiński and Krzysztof Cpałka

Częstochowa University of Technology,
Institute of Computational Intelligence, Poland
{marcin.zalasinski,krzysztof.cpalka}@iisi.pcz.pl

Abstract. Identity verification is one of the biometric issues which may be realized using dynamic signature biometric attribute. One of the methods of signature verification is the method based on partitioning of signature trajectories. In this paper we propose a new method for verification of signature which signals were horizontally partitioned. This method assumes use of all partitions during classification process. Classifier presented in our method is based on the flexible neuro-fuzzy system of the Mamdani type. The algorithm was tested with use of the BioSecure Database (BMDB) distributed by the BioSecure Association.

1 Introduction

On-line signature is a biometric attribute used in identity verification process. In contrast to traditional signatures, on-line signature (called also dynamic signature) contains information about the dynamics of signing. This information are very useful during verification process and increase its accuracy.

In the literature there are many methods of identity verification based on dynamic signature, one of them are methods based on signature partitioning (see [10]). In this paper we propose a new method of classification for dynamic signature which signals are partitioned horizontally. Our method assumes partitioning signatures into four subspaces which are weighted by weights of importance and used during classification process. New classifier presented in this method uses a neuro-fuzzy system (see e.g. [1]-[6], [12]-[18], [22]-[28]).

Relationship between trajectories and other dynamic signals of signature has been described as a very good idea for generation of a discriminative feature set (see [11]). Combination of velocity and pressure with shape makes verification more effective than use of the separated dynamic features. Partitioning allows selection of the most discriminative features of the signature which belong to the user. We assume that all partitions may contain features characteristic for the user but some of them may be less important, so in classification process we use data from all partitions described by weights of importance. The individual steps of the algorithm are detailed below:

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 342–350, 2013.

- **Step 1. Partitioning of signatures.** Signatures are partitioned on the basis of values of velocity and pressure signals. This signals are divided into two parts and then partitioning of the whole signature is performed. Signature elements which time points corresponding to the velocity and pressure signals are assigned to the appropriate partition. After this phase signatures are divided into four parts (two partitions related to the velocity and two partitions related to the pressure).
- **Step 2. Template generation.** In this step templates for each partition are generated. Template contains average values of training signatures signals. This step is performed only during training phase.
- **Step 3. Calculation of distances between signatures and template in each partition.** In training phase similarities between each signature of the user and template are calculated for each partition. The similarities are used for creation of classifier. During test phase similarities are created only for the test signature and they are used by classifier to decide whether the signature is genuine.
- **Step 4. Computation of the weights of importance.** During this step weights of importance for each partition are created. Values of weights are based on mean distance between training signatures and template and also on similarity in distances between training signatures and template. The weights are used by classifier. This step is performed only during training phase.
- **Step 5. Creation decision boundary for each partition.** During this step linear decision boundary in each partition is created (see [31]). This step is performed only during training phase.
- **Step 6. Determination of the fuzzy rules used in classification phase.** Fuzzy rules describe a way of test signature classification. The rules based on the fuzzy sets, which use decision boundaries determined in the step 5. Therefore they may be interpretable.
- **Step 7. Classification.** In this step signature is classified as genuine or forgery. Classification process is performed on the basis of distances between template and sample signature in the partitions. This step is performed only during test phase. In the verification process flexible neuro-fuzzy system of the Mamdani type is used. Each of the antecedents of this classifier is associated with the weight determined in Step 2.

We can see that steps 1-6 are performed during training phase, while steps 1,3,7 are performed during test phase.

Main characteristics of the algorithm may be summarized as follows:

- All partitions described by weights of importance are considered in the classification phase because it is assumed that all partitions may contain some important information about signer.
- Weights of importance are calculated for each partition individually for the signer. High value of weight means that partition correlated with this weight has high value of importance level in the classification phase.

- In the classification phase a classifier based on neuro-fuzzy system of Mamdani type is proposed. These systems are characterized by a very good accuracy. In addition, a selection method of input and output fuzzy sets causes that knowledge stored in the system can be easily interpretable.

This paper is organized into 4 sections. Section 2 contains detailed description of the algorithm. Simulation results are presented in Section 3. Conclusions are drawn in Section 4.

2 Detailed Description of the Algorithm

All training signatures of the i-th signer should be pre-processed by commonly used methods (see e.g. [8]). Pre-processing is used to remove some intra-class variations. All signatures of the i-th signer must be pre-processed with reference to one signature of this user, called base signature.

Next, partitioning of the signature is performed. Velocity and pressure signals are divided into two parts on the basis of them average value from the base signature. After this process horizontal and vertical trajectories of signature are segmented on the basis of generated partitions. The first step of signature partitioning is calculation of average value of signal s (velocity or pressure) of the j-th signature of the i-th signer $\text{avg}_{i,j}^{\{s\}}$, $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$:

$$\text{avg}_{i,j}^{\{s\}} = \frac{1}{K} \sum_{k=1}^{K} s_{i,j,k}, \tag{1}$$

where $s_{i,j,k} \in \{v_{i,j,k}, z_{i,j,k}\}$, $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K$, is signal (velocity or pressure) value of the k-th sample of the j-th signature of the i-th signer.

On the basis of calculated average $\text{avg}_{i,j}^{\{s\}}$, partition $part_{i,j,k}^{\{s\}}$, $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K$, of the k-th sample of the j-th signature of the i-th signer based on s signal (velocity or pressure) can be calculated:

$$part_{i,j,k}^{\{s\}} = \begin{cases} 1 \text{ if } s_{i,j,k} < \text{avg}_{i,j}^{\{s\}} \\ 2 \text{ if } s_{i,j,k} \geq \text{avg}_{i,j}^{\{s\}} \end{cases}, \tag{2}$$

where $s_{i,j,k}$, $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K$, is signal (velocity or pressure) value of the k-th sample of the j-th signature of the i-th signer.

Next, templates of the signatures for each partition are generated. Template $\mathbf{ta}_{p,i}^{\{s\}}$, $p = 1, 2$, $i = 1, 2, \ldots, I$, of the p-th partition of the i-th signer for signatures aligned with use of s signal (v velocity or z pressure) and a trajectory (x or y) is described by the following equation:

$$\mathbf{ta}_{p,i}^{\{s\}} = \left[ta_{p,i,1}^{\{s\}}, ta_{p,i,2}^{\{s\}}, ..., ta_{p,i,k}^{\{s\}} \right], \tag{3}$$

where $ta_{p,i,k}^{\{s\}}$, $p = 1, 2,$, $i = 1, 2, \ldots, I$, $k = 1, 2, \ldots, K$, is template value for the k-th time step of the p-th partition of the i-th signer for signatures aligned with

use of s signal (v velocity or z pressure) and a trajectory (x or y) which is calculated by the formula:

$$ta_{p,i,k}^{\{s\}} = \frac{1}{J}\sum_{j=1}^{J} a_{p,i,j,k}^{\{s\}}, \tag{4}$$

where $a_{p,i,j,k}^{\{s\}}, p = 1, 2, \ldots, PN^{\{s\}}, i = 1, 2, \ldots, I, j = 1, 2, \ldots, J, k = 1, 2, \ldots, K$, is trajectory ($x$ or y) value in the k-th sample of the p-th partition of the j-th signature of the i-th signer.

Next, distances between templates from all partitions and each signature trajectory are calculated. Distance $da_{p,i,j}^{\{s\}}$, $p = 1, 2$, $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$, between template of the p-th partition of s signal (velocity or pressure) of the i-th signer and a trajectory (x or y), and the j-th signature of the i-th signer is described by the following equation:

$$da_{p,i,j}^{\{s\}} = \sqrt{\sum_{k=1}^{K}\left(ta_{p,i,k}^{\{s\}} - a_{p,i,j,k}^{\{s\}}\right)^2}, \tag{5}$$

where $a_{p,i,j,k}^{\{s\}}$, $p = 1, 2$, $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K$, is a a trajectory (x or y) value in the k-th sample of the p-th partition of the j-th signature of the i-th signer.

The next phase of this step is calculation of distances between templates and signatures in two dimensional space. Distance $d_{p,i,j}^{\{s\}}$, $p = 1, 2$, $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$, between the j-th signature trajectory of the i-th signer and template of the i-th signer in the p-th partition of s signal is calculated by the formula:

$$d_{p,i,j}^{\{s\}} = \sqrt{\left(dx_{p,i,j}^{\{s\}}\right)^2 + \left(dy_{p,i,j}^{\{s\}}\right)^2}. \tag{6}$$

In the next step weights of importance for partitions are calculated. The weights are created on the basis of mean distances between signatures and template in partitions and standard deviation of distances in each partition. Mean distance between signatures of the i-th signer and template of the i-th signer in the p-th partition $\bar{d}_{p,i}^{\{s\}}$, $p = 1, 2$, $i = 1, 2, \ldots, I$, related to signal s (v velocity or z pressure) is calculated by the formula:

$$\bar{d}_{p,i}^{\{s\}} = \frac{1}{J}\sum_{j=1}^{J} d_{p,i,j}^{\{s\}}. \tag{7}$$

Standard deviation of signatures $\sigma_{p,i}^{\{s\}}$, $p = 1, 2$, $i = 1, 2, \ldots, I$, of the i-th user from the p-th partition related to signal s (*velocity* or *pressure*) is calculated using the following equation:

$$\sigma_{p,i}^{\{s\}} = \sqrt{\frac{1}{J}\sum_{j=1}^{J}\left(\bar{d}_{p,i}^{\{s\}} - d_{p,i,j}^{\{s\}}\right)^2}. \tag{8}$$

Next, weights of importance are calculated. Weight $w_{p,i}^{'\{s\}}$, $p = 1, 2$, $i = 1, 2, \ldots, I$, of the p-th partition of the i-th user related to signal s (*velocity* or *pressure*) is calculated by the following formula:

$$w_{p,i}^{'\{s\}} = \bar{d}_{p,i}^{\{s\}}\sigma_{p,i}^{\{s\}}. \tag{9}$$

After that weights should be normalized to simplify the classification phase. Weight $w_{p,i}^{\{s\}}$, $p = 1, 2$, $i = 1, 2, \ldots, I$, of the p-th partition of the i-th user related to signal s (*velocity* or *pressure*) is normalized by the following equation:

$$w_{p,i}^{\{s\}} = 1 - \frac{0.9 \cdot w_{p,i}^{'\{s\}}}{\max\left\{w_{1,i}^{'\{s\}}, \ldots, w_{PN^{\{s\}},i}^{'\{s\}}\right\}}. \tag{10}$$

Use of coefficient 0.9 in formula (10) causes that partition with the lowest value of weight of importance is also used in classification process.

Next, selection of location of decision boundary and determination of the value $dlrnmax_{p,i}^{\{s\}}$, $p = 1, 2$, $i = 1, 2, \ldots, I$, $s \in \{v, z\}$ for each partition is performed (see [31]). The determined values have an impact on spacing of fuzzy sets, which represent values $\{\text{low}, \text{high}\}$ assumed by four linguistic variables "the truth of the i-th user signature from p-th partition of s signal" ($p = 1, 2$, $s \in \{v, z\}$).

In the last step signature verification is performed. In this step flexible Mamdani-type neuro-fuzzy system is used (see e.g. [2]-[3], [22]). Our system works on the basis of two fuzzy rules presented as follows:

$$\begin{cases} R^1: \left[\begin{array}{l|l|l} \text{IF}\left(dtst_{1,i}^{\{s\}}\text{is}A_{1,i}^{1\ \{s\}}\right) & w_{1,i}^{\{s\}}\ \text{AND}\left(dtst_{2,i}^{\{s\}}\text{is}A_{2,i}^{1\ \{s\}}\right) & w_{2,i}^{\{s\}} \\ & \text{THEN}y_i\text{is}B^1 & \end{array}\right] \\ R^2: \left[\begin{array}{l|l|l} \text{IF}\left(dtst_{1,i}^{\{s\}}\text{is}A_{1,i}^{2\ \{s\}}\right) & w_{1,i}^{\{s\}}\ \text{AND}\left(dtst_{2,i}^{\{s\}}\text{is}A_{2,i}^{2\ \{s\}}\right) & w_{2,i}^{\{s\}} \\ & \text{THEN}y_i\text{is}B^2 & \end{array}\right] \end{cases}, \tag{11}$$

where

- $dtst_{p,i}^{\{s\}}$, $s \in \{v, z\}$, $p = 1, 2$, $i = 1, 2, \ldots, I$, are input linguistic variables, whose numeric value is a distance between the test signature trajectory of the i-th signer and decision boundary in the p-th partition for signatures aligned with use of s signal.
- $A_{p,i}^{1\ \{s\}}$, $A_{p,i}^{2\ \{s\}}$, $p = 1, 2$, $i = 1, 2, \ldots, I$, are input fuzzy sets related to the signal $s \in \{v, z\}$ shown in Fig. 1. Fuzzy sets $A_{p,i}^{1\ \{s\}}$ and $A_{p,i}^{2\ \{s\}}$ represent values $\{\text{low}, \text{high}\}$ assumed by input linguistic variables $dtst_{p,i}^{\{s\}}$.
- y_i, $i = 1, 2, \ldots, I$, is input linguistic variable interpreted as reliability of signature.

- B^1, B^2 are output fuzzy sets shown in Fig. 1. Fuzzy sets B^1, B^2 represent values $\{\text{low}, \text{high}\}$ assumed by output linguistic variable determining the reliability of signature.
- $w_{p,i}^{\{s\}}$, $p = 1, 2$, $i = 1, 2, \ldots, I$, $s \in \{v, z\}$, are weights of the p-th partition of the i-th user related to signal s.

The rules from (7) can be written in a simpler form:

$$\begin{cases} R^1 : \left[\begin{array}{c} \text{IF } \left(dtst_{1,i}^{\{s\}} \text{is low}\right) \left| w_{1,i}^{\{s\}} \text{ AND } \left(dtst_{2,i}^{\{s\}} \text{is low}\right) \right| w_{2,i}^{\{s\}} \\ \text{THEN} y_i \text{is low} \end{array} \right] \\ R^2 : \left[\begin{array}{c} \text{IF } \left(dtst_{1,i}^{\{s\}} \text{is high}\right) \left| w_{1,i}^{\{s\}} \text{ AND } \left(dtst_{2,i}^{\{s\}} \text{is high}\right) \right| w_{2,i}^{\{s\}} \\ \text{THEN} y_i \text{is high} \end{array} \right] \end{cases}. \tag{12}$$

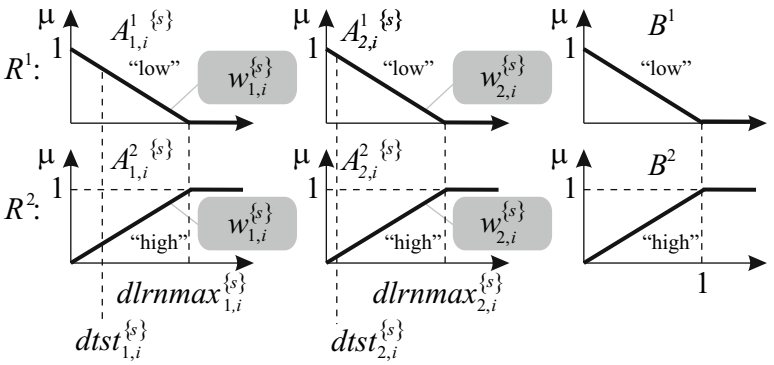


Fig. 1. Input and output fuzzy sets of the flexible neuro-fuzzy system of the Mamdani type for signature verification

A signature is considered true if the following assumption is satisfied:

$$\bar{y}_i = \frac{T^* \left\{ \mu_{A_{1,i}^{2\ \{s\}}} \left(dtst_{1,i}^{\{s\}}\right), \mu_{A_{2,i}^{2\ \{s\}}} \left(dtst_{2,i}^{\{s\}}\right); w_{1,i}^{\{s\}}, w_{2,i}^{\{s\}} \right\}}{\left(\begin{array}{c} T^* \left\{ \mu_{A_{1,i}^{2\ \{s\}}} \left(dtst_{1,i}^{\{s\}}\right), \mu_{A_{2,i}^{2\ \{s\}}} \left(dtst_{2,i}^{\{s\}}\right); w_{1,i}^{\{s\}}, w_{2,i}^{\{s\}} \right\} + \\ T^* \left\{ \mu_{A_{1,i}^{1\ \{s\}}} \left(dtst_{1,i}^{\{s\}}\right), \mu_{A_{2,i}^{1\ \{s\}}} \left(dtst_{2,i}^{\{s\}}\right); w_{1,i}^{\{s\}}, w_{2,i}^{\{s\}} \right\} \end{array} \right)} > cth_i, \tag{13}$$

where

- $S^* \{\cdot\}$ is a weighted t-conorm (see [2]).
- $\bar{y}_i$, $i = 1, 2, \ldots, I$, is the value of the output signal of applied neuro-fuzzy system described by rules (11). Detailed description of the system can be found in [2]. Formula (13) is the result of the general relationship describing the transformation of the input signal of Mamdani-type system.

- $cth_i \in [0,1]$ - coefficient determined experimentally during training phase for each user to eliminate disproportion between FAR and FRR error (see [29]). The parameters $cth_i \in [0,1]$, computed individually for the i-th user, $i = 1, 2, \ldots, I$, are used during verification process in the test phase.

In future research we plan to use probabilistic neural networks for classification of dynamic signature ([7], [19]-[21]).

3 Simulation Results

The simulation was performed with use of BioSecure Database (BMDB) distributed by the BioSecure Association. We used DS2 Signature database which contains signatures of 210 users. The signatures was acquired in two sessions using the digitizing tablet. Each session contains 15 genuine signatures and 10 skilled forgeries per person. Our test was performed five times for all signers. During training phase 5 genuine signatures of each signer were used. During test phase 10 genuine signatures and 10 forged signatures of each signer were used. The test was performed by the authorial testing environment implemented in C# language. We also implemented the method presented in [8] to compare the results. Table 1 contains simulation results described as values of FAR (False Acceptance Rate) and FRR (False Rejection Rate), which are commonly use in biometrics (see e.g. [9]). It should be noted, that the method proposed in this paper achieves the best results.

Table 1. Results of simulation performed by our system

Method	**Average FAR**	**Average FRR**	**Average error**
Khan et al. [8]	2.94 %	6.90 %	4.92 %
Our method	**2.92 %**	**4.46 %**	**3.69 %**

4 Conclusions

In this paper a new method for the on-line signature verification based on horizontal partitioning is presented. The signature is divided into four parts which are used during classification process. The method assumes use of classifier based on the Mamdani type neuro-fuzzy system which is characterized by a very good accuracy and ease of interpretation of the collected knowledge. The achieved accuracy of our simulations performed using BioSecure database proves the correctness of the proposed assumptions.

Acknowledgment. The project was financed by the National Science Center on the basis of the decision number DEC-2011/01/N/ST6/06964.

References

1. Cpałka, K., Rutkowski, L.: A new method for designing and reduction of neuro-fuzzy systems. In: IEEE Int. Conference on Fuzzy Systems, pp. 1851–1857 (2006)
2. Cpalka, K.: A method for designing flexible neuro-fuzzy systems. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 212–219. Springer, Heidelberg (2006)
3. Cpałka, K., Rutkowski, L.: Neuro-fuzzy structures for pattern classification. WSEAS Trans. on Computers, 697–688 (2005)
4. Cpałka, K., Rutkowski, L.: Neuro-fuzzy systems derived from quasi-triangular norms. In: IEEE International Conference on Fuzzy Systems, vol. 2, pp. 1031–1036 (2004)
5. Cpałka, K.: On evolutionary designing and learning of flexible neuro-fuzzy structures for nonlinear classification. Nonlinear Analysis Series A: Theory, Methods & Applications 71 (2009)
6. Gabryel, M., Rutkowski, L.: Evolutionary methods for designing neuro-fuzzy modular systems combined by bagging algorithm. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 398–404. Springer, Heidelberg (2008)
7. Greblicki, W., Rutkowski, L.: Density-free Bayes risk consistency of nonparametric pattern recognition procedures. Proceedings of the IEEE 69(4), 482–483 (1981)
8. Ibrahim, M.T., Khan, M.A., Alimgeer, K.S., Khan, M.K., Taj, I.A., Guan, L.: Velocity and pressure-based partitions of horizontal and vertical trajectories for on-line signature verification. Pattern Recognition 43 (2010)
9. Jain, A.K., Ross, A.: Introduction to Biometrics. In: Jain, A.K., Flynn, P., Ross, A.A. (eds.) Handbook of Biometrics. Springer (2008)
10. Khan, M.A.U., Khan, M.K., Khan, M.A.: Velocity-image model for online signature verification. IEEETrans. Image Process 15 (2006)
11. Khan, M.K., Khan, M.A., Khan, M.A.U., Lee, S.: Signature verification using velocity-based directional filter bank. In: IEEE Asia Pacific Conference on Circuits and Systems, APCCAS, pp. 231–234 (2006)
12. Korytkowski, M., Gabryel, M., Rutkowski, L., Drozda, S.: Evolutionary methods to create interpretable modular system. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 405–413. Springer, Heidelberg (2008)
13. Korytkowski, M., Rutkowski, L., Scherer, R.: On combining backpropagation with boosting. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN), Vancouver, pp. 1274–1277 (2005)
14. Li, X., Er, M.J., Lim, B.S., et al.: Fuzzy Regression Modeling for Tool Performance Prediction and Degradation Detection. International Journal of Neural Systems 20(5), 405–419 (2010)
15. Nowicki, R., Pokropińska, A.: Information criterions applied to neuro-fuzzy architectures design. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 332–337. Springer, Heidelberg (2004)
16. Nowicki, R., Scherer, R., Rutkowski, L.: A method for learning of hierarchical fuzzy systems. In: Sincak, P., Vascak, J., Kvasnicka, V., Pospichal, J. (eds.) Intelligent Technologies - Theory and Applications, pp. 124–129. IOS Press (2002)

17. Przybył, A., Cpałka, K.: A new method to construct of interpretable models of dynamic systems. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 697–705. Springer, Heidelberg (2012)
18. Rutkowska, D., Nowicki, R., Rutkowski, L.: Neuro-fuzzy architectures with various implication operators. In: Sincak, P., et al. (eds.) The State of the Art in Computational Intelligence, pp. 214–219 (2000)
19. Rutkowski, L.: Sequential estimates of probability densities by orthogonal series and their application in pattern classification. IEEE Transactions on Systems, Man, and Cybernetics SMC-10(12), 918–920 (1980)
20. Rutkowski, L.: On Bayes risk consistent pattern recognition procedures in a quasi-stationary environment. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-4(1), 84–87 (1982)
21. Rutkowski, L.: Adaptive probabilistic neural-networks for pattern classification in time-varying environment. IEEE Transactions on Neural Networks 15, 811–827 (2004), Rutkowski, L.: Generalized regression neural networks in time-varying environment. IEEE Transactions on Neural Networks 15, 576–596 (2004)
22. Rutkowski, L.: Computational intelligence. Springer (2007)
23. Rutkowski, L.: Cpałka K, Flexible weighted neuro-fuzzy systems, Neural Information Processing. In: Proceedings of the 9th International Conference on ICONIP 2002, pp. 1857–1861 (2002)
24. Rutkowski, L., Przybył, A., Cpałka, K., Er, M.J.: Online Speed Profile Generation for Industrial Machine Tool Based on Neuro-fuzzy Approach. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 645–650. Springer, Heidelberg (2010)
25. Rutkowski, L., Przybył, A., Cpałka, K.: Novel Online Speed Profile Generation for Industrial Machine Tool Based on Flexible Neuro-Fuzzy Approximation. IEEE Transactions on Industrial Electronics 59(2), 1238–1247 (2012)
26. Scherer, R., Rutkowski, L.: A fuzzy relational system with linguistic antecedent certainty factors. In: 6th International Conference on Neural Networks and Soft Computing, Zakopane, Poland. Advances In Soft Computing, pp. 563–569 (2003)
27. Scherer, R., Rutkowski, L.: Connectionist fuzzy relational systems. In: Halgamuge, S.K., Wang, L. (eds.) Computational Intelligence for Modelling and Prediction. SCI, vol. 2, pp. 35–47. Springer, Heidelberg (2005)
28. Scherer, R., Rutkowski, L.: Neuro-fuzzy relational classifiers. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 376–380. Springer, Heidelberg (2004)
29. Yeung, D.-Y., Chang, H., Xiong, Y., George, S.E., Kashi, R.S., Matsumoto, T., Rigoll, G.: SVC2004: First International Signature Verification Competition. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, pp. 16–22. Springer, Heidelberg (2004)
30. Zalasiński, M., Cpałka, K.: A new method of on-line signature verification using a flexible fuzzy one-class classifier. In: Selected Topics in Computer Science Applications, pp. 38–53. EXIT (2011)
31. Zalasiński, M., Cpałka, K.: Novel algorithm for the on-line signature verification. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 362–367. Springer, Heidelberg (2012)

Adaptive Critic Designs in Control of Robots Formation in Unknown Environment

Zenon Hendzel, Andrzej Burghardt, and Marcin Szuster

Rzeszow University of Technology
Department of Applied Mechanics and Robotics
8 Powstancow Warszawy St., 35-959 Rzeszow, Poland
{zenhen,andrzejb,mszuster}@prz.edu.pl

Abstract. In the presented article a new approach to a collision-free trajectory generating for a wheeled mobile robots formation with Adaptive Critic Designs and Fuzzy Logic algorithm, is proposed. The presented hierarchical control system consists of a trajectory generating algorithm based on a conception of reactive navigation of the wheeled mobile robots formation in the unknown 2D environment, a control system that generates individual trajectories for all agents in formation, and agents tracking control systems. A strategy of reactive navigation is developed including two main behaviours: a obstacle avoiding behaviour and a goal-seeking behaviour, realised in a form of Adaptive Critic Design algorithms. These individual behaviours are combined using two approaches: cooperative connection approach and the fuzzy combiner, that determines influence of the individual behaviours on the trajectory generation process, according to the environment conditions. Computer simulations have been conducted to illustrate the process of path planning in different environment conditions.

Keywords: Adaptive Critic Design, Behavioural Control, Fuzzy System, Neural Network, Reactive Navigation, Mobile Robots Formation.

1 Introduction

The development of mobile robotics in recent years allowed to increase area of its applications. Simultaneously it made realization of more complex tasks possible and involved necessity of more complicated control systems development. Increase of the wheeled mobile robots (WMRs) constructions complexity, quantity of information received from the environment, and performance of microprocessors, allowed to design control systems capable of generating a WMRs motion trajectory in a real time and modifying it according to the environment conditions, e.g. position of obstacles. The main aim of the work concerns a control problem of a wheeled mobile robots formation (WMRF) for military applications such as: movement in a defined order, tracing, border control. Other civil highly developed applications are warehouses monitoring, map making, transport and space exploration [1]. Robots control is strictly connected with the type

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 351–362, 2013.

of a realised task. In patrol, military and inspection tasks agents can change its position in a structure of the WMRF. This can be achieved using formation control algorithms, in which all agents are tracing the leader [11, 15]. Another type of formation task is e.g. large scale object transport, where the formation has to remain in the same configuration with no possibility of any changes in the sequence of robots. These type of tasks are developed using the virtual structure algorithms [3]. Methods of the WMRF control inspired by live organisms, are called behavioural methods [9, 10]. Artificial Intelligence (AI) algorithms, as Neural Networks (NNs) or Fuzzy Logic (FL) systems, are widely use to solve this kind of problems. The development of AI methods allowed to apply Bellman's Dynamic Programming (DP) idea in a form of Approximate Dynamic Programming (ADP) algorithms, also known as Adaptive Critic Designs (ACDs) [12–14]. ACDs make generating of the sub-optimal control law in forward processes possible.

In presented article a new approach to a collision-free trajectory generating for the WMRF, with usage of ACD algorithms, is proposed. The designed hierarchical control system consists of the trajectory generator, based on ACDs in Action Dependant Heuristic Dynamic Programming (ADHDP) configuration, that generate behavioural control signals in the goal seeking (GS) and the obstacle avoiding (OA) tasks. Two approaches to combination of behavioural control signals in GS and OA tasks were developed. The first one is called the cooperative connection of behavioural control signals, in this approach influence of two simple behaviours on the trajectory generation control signals is constant. In the second approach influence of elementary behaviours changes fluently according to the fuzzy system rules base, taking into account the environment conditions, as distances to the goal and nearest obstacles. This approach guarantees generation of successful collision-free trajectories in the complex task of goal seeking with obstacle avoiding, and realisation of these trajectories using the individual tracking control systems with ACDs in Dual Heuristic Programming (DHP) configuration for all agents in the formation. The results of researches presented in the article continue authors' earlier works related to the path planning processes in the unknown 2D environment [2, 8], the WMRF control [5], and the tracking control of the WMR using ADP methods [6, 7]. The paper is organised in the following way: the first section includes a short introduction into the WMRF path planning problems, the second section presents the WMR and the WMRF. Next section includes the description of the proposed hierarchical control system, with the trajectory generator, formation control algorithm and the tracking control system. In the following sections there are presented results of numerical tests and summary of the research project.

2 AmigoBot Mobile Robots

In the WMRF were used models of AmigoBot WMRs (Fig. 1). The non-holonomic Amigobot consists of two driving wheels (1 and 2), a frame 4 and the third, free rolling castor wheel 3 (Fig. 1(b)). Its movement is analysed in the xy plane [4].

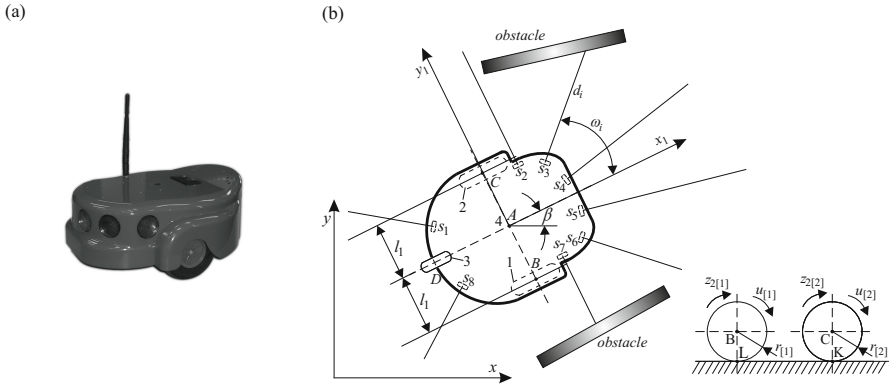


Fig. 1. (a) Mobile robot AmigoBot, (b) model of the wheeled mobile robot

2.1 Kinematics of the Wheeled Mobile Robot

On the basis of the assumed joint coordinates $\boldsymbol{q} = [x_A, y_A, \beta, \alpha_1, \alpha_2]^T$ a non-holonomic equations of the WMR takes the form [4]

$$\boldsymbol{J}(\boldsymbol{q})\,\dot{\boldsymbol{q}} = \boldsymbol{0}\,, \tag{1}$$

where

$$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} 1 & 0 & l_1\cos(\beta) & -r\cos(\beta) & 0 \\ 0 & 1 & -l_1\sin(\beta) & 0 & -r\sin(\beta) \end{bmatrix}\,, \tag{2}$$

and l_1 – dimension, that derives from the WMR geometry, $r_{[1]} = r_{[2]} = r$ – radius of the driving wheel, β – the angle of the WMR's frame self turn.

The kinematics of the WMR is described by the equation

$$\begin{bmatrix} \dot{x}_A \\ \dot{y}_A \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} v_A^* \cos(\beta) & 0 \\ v_A^* \sin(\beta) & 0 \\ 0 & \dot{\beta}^* \end{bmatrix} \begin{bmatrix} u_v \\ u_\beta \end{bmatrix}\,, \tag{3}$$

where x_A, y_A – coordinates of the point A of the WMR's frame, v_A^* – a maximal defined velocity of the point A, $\dot{\beta}^*$ – a maximal defined angular velocity of the self turn of the WMR's frame, $\boldsymbol{u}_B = [u_v, u_\beta]^T$ – control signals of the trajectory generator. The presented WMRs AmigoBot can be used to create the formation with localisation of individual robots determined by the type of a realised task. Typical examples of formations are: a column, a triangle, a diamond and a line.

2.2 Dynamics of the Wheeled Mobile Robot

The dynamics of the WMR was modelled using Maggie's mathematical formalism [4]. Using Euler's derivative approximation and the state vector $\boldsymbol{z}_{\{k\}} = \left[\boldsymbol{z}_{1\{k\}}^T, \boldsymbol{z}_{2\{k\}}^T\right]^T$, where $\boldsymbol{z}_{2\{k\}}$ corresponds to the vector of continuous angular

velocities $\dot{\boldsymbol{\alpha}} = \left[\dot{\alpha}_{[1]}, \dot{\alpha}_{[2]}\right]^T$, a discrete notation of the WMR dynamics can be written as

$$\begin{aligned} \boldsymbol{z}_{1\{k+1\}} &= \boldsymbol{z}_{1\{k\}} + \boldsymbol{z}_{2\{k\}} h \,, \\ \boldsymbol{z}_{2\{k+1\}} &= -\boldsymbol{M}_R^{-1} \left[\boldsymbol{C}\left(\boldsymbol{z}_{2\{k\}}\right) \boldsymbol{z}_{2\{k\}} + \boldsymbol{F}\left(\boldsymbol{z}_{2\{k\}}\right) + \boldsymbol{\tau}_{d\{k\}} - \boldsymbol{u}_{\{k\}}\right] h + \boldsymbol{z}_{2\{k\}} \,, \end{aligned} \tag{4}$$

where $\boldsymbol{M}_R$, $\boldsymbol{C}\left(\boldsymbol{z}_{2\{k\}}\right)$, $\boldsymbol{F}\left(\boldsymbol{z}_{2\{k\}}\right)$ – matrixes and vectors that derive from the WMR dynamics, $\boldsymbol{\tau}_{d\{k\}}$ – the vector of bounded disturbances, $\boldsymbol{u}_{\{k\}}$ – the tracking control signal, h – a time discretisation parameter, k – an index of iteration steps, $k = 0, \ldots, n$, n – number of iteration steps.

The continuous dynamics model of the WMR was described in detail in [4], the closed loop system used in the tracking control system synthesis, was described in detail in [6, 7].

2.3 Wheeled Mobile Robots Formation

The scheme of the WMRF is shown in Fig. 2.

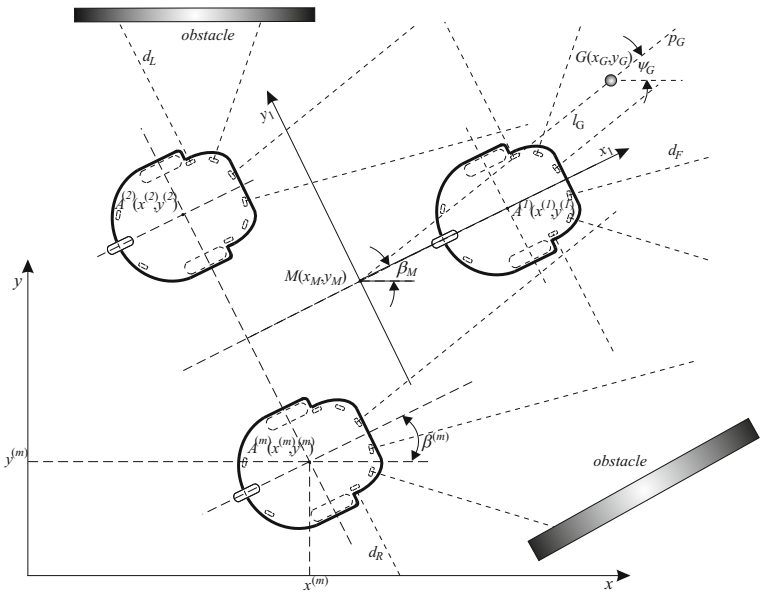


Fig. 2. Scheme of the wheeled mobile robots formation

In the article was used the WMRF that consists of $m = 3$ agents, that form a virtual structure of the equilateral triangle. The WMR is equipped with eight ultrasonic range finders, for obstacles detection. Range finders of the j-th WMR are denoted by $s_1^{(j)}, \ldots, s_8^{(j)}$, and measurements by $d_1^{(j)}\left(s_1^{(j)}\right), \ldots, d_8^{(j)}\left(s_8^{(j)}\right)$. Not all range finder signals are used by the trajectory generator, they are grouped

to obtain normalised distances to the obstacles in the front of the WMRF (d_F^*), on the right side (d_R^*) and on the left side (d_L^*) of the WMRF. The way of normalising distances to the obstacles is described in the section describing the trajectory generator. $A^{(j)}\left(x^{(j)}, y^{(j)}\right)$ are coordinates of the j-th WMR's point A. The point M is a central point of the WMRF, β_M is the angle of the WMRF's virtual structure self turn.

3 Hierarchical Control System

The task of reaching the goal by the WMRF in the unknown environment with static obstacles was realised using the hierarchical control system that consists of three layers: the trajectory generator, the formation control system and individual tracking control systems for all three agents. The trajectory generator and the tracking control systems were both build using ACDs. The scheme of the hierarchical control system is shown in Fig. 3.

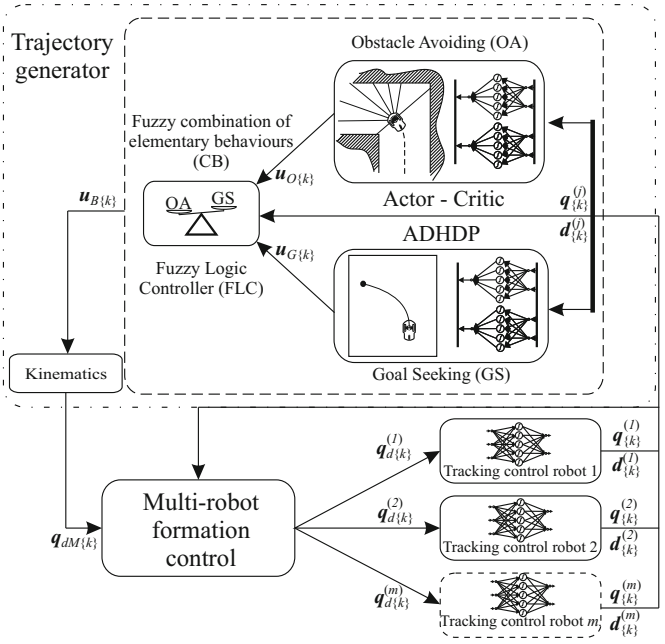


Fig. 3. Scheme of the hierarchical control system

Trajectory Generator. In the presented trajectory generator a strategy of reactive navigation is developed including two main behaviours: OA and GS [1, 2, 5, 8]. These simple, individual behaviours are combined by the constant coefficient in the cooperative connection approach or by the fuzzy combiner of behaviours, that determines influence of the individual behaviours on the trajectory generation process, according to the environment conditions.

Behavioural Control System in the Goal-Seeking Behaviour. Behavioural control signals in GS behaviour are generated using discrete ACD algorithms in ADHDP configuration, in detail described in [6].

The objective of the ACD algorithm is to determine the sub-optimal control law, that minimizes the value function $V_{\{k\}}\left(\boldsymbol{x}_{\{k\}}, \boldsymbol{u}_{\{k\}}\right)$ [10,11,12], which is the function of a state $\boldsymbol{x}_{\{k\}}$ and a control $\boldsymbol{u}_{\{k\}}$ in general case

$$V_{\{k\}}\left(\boldsymbol{x}_{\{k\}}, \boldsymbol{u}_{\{k\}}\right) = \sum_{k=0}^{n} \gamma^k L_{C\{k\}}\left(\boldsymbol{x}_{\{k\}}, \boldsymbol{u}_{\{k\}}\right), \tag{5}$$

where γ – a discount factor $(0 \leq \gamma \leq 1)$, $L_{C\{k\}}\left(\boldsymbol{x}_{\{k\}}, \boldsymbol{u}_{\{k\}}\right)$ – a local cost in a step k.

The generated velocity error $e_{Gv\{k\}}$ and the angle of the WMRF turn error $e_{G\dot{\beta}\{k\}}$ for the GS behaviour are defined in the form

$$\begin{aligned} e_{Gv\{k\}} &= f\left(d^*_{G\{k\}}\right) - \tfrac{v_{M\{k\}}}{v^*_M}\ , \\ e_{G\dot{\beta}\{k\}} &= \varphi_{G\{k\}} - \beta_{M\{k\}}\ , \end{aligned} \tag{6}$$

where $f(.)$ – a sigmoidal unipolar function, $d^*_{G\{k\}} \in < 0, 1 >$ – the normalised distance to the goal G, $d^*_{G\{k\}} = d_{G\{k\}}/d_{Gmx}$, d_{Gmx} – the maximal distance to the goal G, $v_{M\{k\}}$ – a realised velocity of the point M of the WMRF, v^*_M – a maximal defined velocity of the point M, $\varphi_{G\{k\}}$ - an angle between the axis of the WMRF's virtual structure and the straight line p_G.

The local costs $L_{CGv\{k\}}\left(e_{Gv\{k\}}, u_{Gv\{k\}}\right)$, and $L_{CG\dot{\beta}\{k\}}\left(e_{G\dot{\beta}\{k\}}, u_{G\dot{\beta}\{k\}}\right)$ were assumed in the forms

$$\begin{aligned} L_{CGv\{k\}}\left(e_{Gv\{k\}}, u_{Gv\{k\}}\right) &= \tfrac{1}{2} R_{Gv} e^2_{Gv\{k\}} + \tfrac{1}{2} Q_{Gv} u^2_{Gv\{k\}}\ , \\ L_{CG\dot{\beta}\{k\}}\left(e_{G\dot{\beta}\{k\}}, u_{G\dot{\beta}\{k\}}\right) &= \tfrac{1}{2} R_{G\dot{\beta}} e^2_{G\dot{\beta}\{k\}} + \tfrac{1}{2} Q_{G\dot{\beta}} u^2_{G\dot{\beta}\{k\}}\ , \end{aligned} \tag{7}$$

where R_{Gv}, Q_{Gv}, $R_{G\dot{\beta}}$, $Q_{G\dot{\beta}}$ – positive constants, $u_{Gv\{k\}}$, $u_{G\dot{\beta}\{k\}}$ – the overall behavioural control signals, that consist of control signals generated by actor NNs $\boldsymbol{u}_{GA\{k\}} = \left[u_{GAv\{k\}}, u_{GA\dot{\beta}\{k\}}\right]^T$, and proportional (P) controller signal

$$\boldsymbol{u}_{G\{k\}} = \boldsymbol{u}_{\mathrm{GA}\{k\}} + \boldsymbol{u}_{GP\{k\}}\ , \tag{8}$$

where $\boldsymbol{u}_{GP\{k\}} = \boldsymbol{K}_{GP}\left[e_{Gv\{k\}}, e_{G\dot{\beta}\{k\}}\right]^T$, $\boldsymbol{K}_{GP}$ – a positive defined, fixed diagonal matrix.

ACD structures are classified as Reinforcement Learning (RL) methods, where algorithms search for the optimal control law by exploring acceptable control laws and states of the system, and exploiting obtained strategies. Use of the proportional controller in the presented behavioural control system is an innovative approach, that limits exploration by prompting the ACD structure proper control signal at the beginning of the NNs' weights adaptation process, what prevents from the trial and error learning.

The behavioural control signals $\boldsymbol{u}_{GA\{k\}}$ in the GS task are generated by two ADHDP actor-critic structures, composed of;

– critic, that estimates the sub-optimal value function $V_{Gv\{k\}}\left(e_{Gv\{k\}}, u_{Gv\{k\}}\right)$ and $V_{G\dot{\beta}\{k\}}\left(e_{G\dot{\beta}\{k\}}, u_{G\dot{\beta}\{k\}}\right)$, and is realised in the form of Random Vector Functional Link (RVFL) NNs with output signals

$$\begin{aligned} \hat{V}_{Gv\{k\}}\left(e_{Gv\{k\}}, u_{Gv\{k\}}\right) &= \boldsymbol{W}_{GC1\{k\}}^{T}\boldsymbol{S}\left(\boldsymbol{x}_{GCv\{k\}}\right) , \\ \hat{V}_{G\dot{\beta}\{k\}}\left(e_{G\dot{\beta}\{k\}}, u_{G\dot{\beta}\{k\}}\right) &= \boldsymbol{W}_{GC2\{k\}}^{T}\boldsymbol{S}\left(\boldsymbol{x}_{GC\dot{\beta}\{k\}}\right) , \end{aligned} \tag{9}$$

where $\boldsymbol{W}_{GC1\{k\}}, \boldsymbol{W}_{GC2\{k\}}$ – vectors of output-layer weights, $\boldsymbol{S}(.)$ - a vector of sigmoidal bipolar neurons activation functions, $\boldsymbol{x}_{GCv\{k\}}, \boldsymbol{x}_{GC\dot{\beta}\{k\}}$ – NNs' input vectors. Schematic structure of the critic's RVFL $\hat{V}_{Gv\{k\}}\left(e_{Gv\{k\}}, u_{Gv\{k\}}\right)$ NN is shown in Fig. 4, where $\boldsymbol{D}_{GC}$ – a matrix of fixed input weights, chosen randomly in the NN's initialization process, J – number of inputs, N – number of neurons. Critics' weights are adapted by the gradient method of the Temporal Difference errors in the forms

$$\begin{aligned} e_{GCv\{k\}} &= L_{GCv\{k\}}\left(e_{Gv\{k\}}, u_{Gv\{k\}}\right) + \gamma\hat{V}_{Gv\{k+1\}}\left(\boldsymbol{x}_{GCv\{k+1\}}, \boldsymbol{W}_{GC1\{k\}}\right) + \\ &\quad -\hat{V}_{Gv\{k\}}\left(\boldsymbol{x}_{Gv\{k\}}, \boldsymbol{W}_{GC1\{k\}}\right) , \\ e_{GC\dot{\beta}\{k\}} &= L_{CG\dot{\beta}\{k\}}\left(e_{G\dot{\beta}\{k\}}, u_{G\dot{\beta}\{k\}}\right) + \gamma\hat{V}_{G\dot{\beta}\{k+1\}}\left(\boldsymbol{x}_{G\dot{\beta}\{k+1\}}, \boldsymbol{W}_{GC2\{k\}}\right) + \\ &\quad -\hat{V}_{\dot{\beta}\{k\}}\left(\boldsymbol{x}_{G\dot{\beta}\{k\}}, \boldsymbol{W}_{GC2\{k\}}\right) . \end{aligned} \tag{10}$$

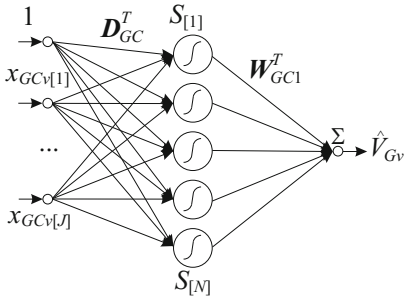


Fig. 4. Scheme of the critic's RVFL $\hat{V}_{Gv\{k\}}\left(e_{Gv\{k\}}, u_{Gv\{k\}}\right)$ NN

– actor, that generates the sub-optimal control law $u_{GAv\{k\}}$ and $u_{GA\dot{\beta}\{k\}}$, is realised in the form of RVFL NNs with output signals

$$\begin{aligned} u_{GAv\{k\}}\left(\boldsymbol{x}_{GAv\{k\}}, \boldsymbol{W}_{GA1\{k\}}\right) &= \boldsymbol{W}_{GA1\{k\}}^{T}\boldsymbol{S}\left(\boldsymbol{x}_{GAv\{k\}}\right) , \\ u_{GA\dot{\beta}\{k\}}\left(\boldsymbol{x}_{GA\dot{\beta}\{k\}}, \boldsymbol{W}_{GA2\dot{\beta}\{k\}}\right) &= \boldsymbol{W}_{GA2\dot{\beta}\{k\}}^{T}\boldsymbol{S}\left(\boldsymbol{x}_{GA\dot{\beta}\{k\}}\right) , \end{aligned} \tag{11}$$

and theirs weights are adapted by the back propagation method of errors

$$e_{GAv\{k\}} = \frac{\partial L_{CGv\{k\}}\left(e_{Gv\{k\}}, u_{Gv\{k\}}\right)}{\partial u_{Gv\{k\}}} + \gamma \frac{\partial \hat{V}_{Gv\{k+1\}}\left(\boldsymbol{x}_{Gv\{k+1\}}, \boldsymbol{W}_{GC1\{k\}}\right)}{\partial u_{Gv\{k\}}} ,$$
$$e_{GA\dot{\beta}\{k\}} = \frac{\partial L_{CG\dot{\beta}\{k\}}\left(e_{G\dot{\beta}\{k\}}, u_{G\dot{\beta}\{k\}}\right)}{\partial u_{G\dot{\beta}\{k\}}} + \gamma \frac{\partial \hat{V}_{G\dot{\beta}\{k+1\}}\left(\boldsymbol{x}_{G\dot{\beta}\{k+1\}}, \boldsymbol{W}_{GC2\dot{\beta}\{k\}}\right)}{\partial u_{G\dot{\beta}\{k\}}} . \quad (12)$$

In the behavioural control systems were used RVFL NNs with fixed input-layer weights, randomly chosen in the initialization process, set to zero initial output-layer weights and neurons with sigmoidal bipolar activation functions. Each NN had eight neuron activation functions.

Behavioural Control System in the Obstacle Avoiding Behaviour. The control system in the OA task was built in the similar way, that in the GS task and was described in details in [8].

Fuzzy Combiner of Behaviours. We used the Takagi-Sugeno FL model, with triangular or trapezoidal affiliation functions to fuzzy sets. The FL controller contains the rules base, that consists of $m = 25$ rules in the form

$$R_B^m : \text{IF } \left(d^*_{G\{k\}} \text{ is dS}\right) \text{ AND } \left(d^*_{O\{k\}} \text{ is dS}\right) \text{ THEN } a_{B\{k\}} \text{ is aB} , \quad (13)$$

where $d^*_{O\{k\}} in < 0, 1 >$ – the normalised distance to the obstacle, calculated on the basis of the distance measurements, $a_{B\{k\}}$ – the combination of individual behaviours control signal, "lS", "dS", "aB" - linguistic labels of affiliation functions to the fuzzy sets. Scheme of the rules base is shown in Fig. 5, where linguistic labels of particular affiliation functions to the fuzzy sets are: "WS0"– very small, near zero, "WS" – very small, "S" – small, "M" – medium, "B" – big, "WB" – very big, "WB1" – very big, near one.

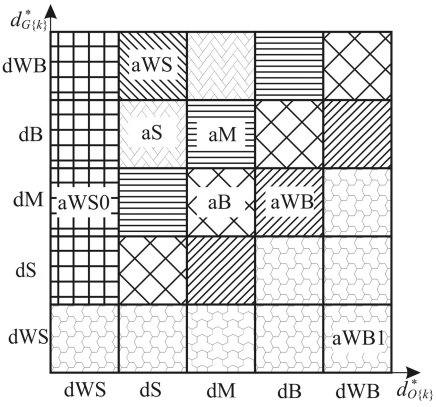


Fig. 5. Scheme of the rules base

The trajectory generation system generates the control signal $\boldsymbol{u}_{B\{k\}}$, on the basis of control signals generated for the individual behaviours; goal-seeking $\boldsymbol{u}_{G\{k\}} = \left[u_{Gv\{k\}}, u_{G\dot{\beta}\{k\}}\right]^T$, and obstacle avoiding $\boldsymbol{u}_{O\{k\}} = \left[u_{Ov\{k\}}, u_{O\dot{\beta}\{k\}}\right]^T$, according to equation

$$\boldsymbol{u}_{B\{k\}} = a_{B\{k\}}\boldsymbol{u}_{G\{k\}} + \left(1 - a_{B\{k\}}\right) a_{0\{k\}} \ , \tag{14}$$

The control signal of the trajectory generator $\boldsymbol{u}_{B\{k\}}$ is used by the formation control system to generate the trajectory of the WMRF.

Robots Formation Control System. The algorithm of formation control realises the idea of using the virtual structure with defined central point of the formation $M\left(x_M, y_M\right)$ and orientation of the virtual structure β_M. It generates desired trajectories for individual agents and was described in detail in [5].

Tracking Control System. The tracking control systems for individual agents generate control signals that make possible to realise trajectories generated by higher layers of the hierarchical control system. The individual tracking control system consist of the ACD in the Dual Heuristic Dynamic Programming (DHP) configuration, PD controller and the supervisory therm. It was described in [7].

4 Experiment Results

Numerical tests of the hierarchical control system was realised by a series of simulations of the WMRF movement in the numerical environment, using Matlab/Simulink software. In this section, for the sake of simplicity index k is omitted, h=0.01 [s].

On the basis of range finder signals the proposed control system generated the collision free paths of the point M of the virtual structure from the starting point to the goals $G_1\left(6.5, 7.5\right)$ and $G_2\left(9.5, 8.0\right)$. Two approaches to combination of behavioural control signals of GS and OA tasks were compared. In the first approach, called the cooperative connection of elementary tasks, influence of two simple behaviours on the trajectory generation control signals is constant, a_B=const. In the second one influence of these behaviours changes fluently according to the fuzzy system rules base, taking into account the environment conditions, as distances to the goal and nearest obstacles. The environment maps with trajectories of the WMR in formation, positions of obstacles and the destination in points $G_1\left(6.5, 7.5\right)$ and $G_2\left(9.5, 8.0\right)$, are shown in Fig. 6.

In the figure the start position of the WMRs are marked by triangles, the goal is marked by the "X". Trajectories generated using the cooperative connection of elementary tasks are shown in Fig. 6(a), and trajectories obtained by the use of the fuzzy control system are shown in Fig. 6(b).

The map of the environment was projected in the way, that none of the behavioural control system in the OA or the GS task is able to generate the successive path itself, it is possible on the basis of the control signal generated by the presented algorithm with cooperation of the individual behaviours.

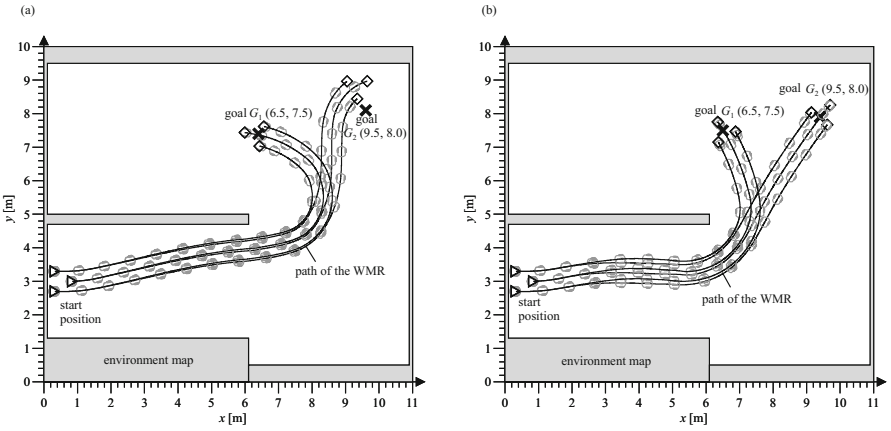


Fig. 6. The environment map with the path of the WMRF to the goal $G_1\,(6.5, 7.5)$ and $G_2\,(9.5, 8.0)$, obtained using: (a) the cooperative connection approach (a_B=const.) (test 1–G_1 and 2–G_2), (b) the fuzzy logic control system (test 3–G_1 and 4–G_2).

In the further part of the article are presented result of the numerical test obtained for the numerical test 4 with the goal $G_2\,(9.5, 8.0)$ using the trajectory generator with fuzzy logic control system for soft switching of behavioural control signals. On the basis of range finders signals the proposed control system generated the collision free path of the point M of the WMRF from the starting point to the goal $G_2\,(9.5, 8.0)$, with the FL combination of behaviours control signal a_B shown in Fig. 7(a) and the trajectory generator control signals u_{Bv} and $u_{B\dot{\beta}}$, shown in Fig. 7(b). The control signals u_{Bv} and $u_{B\dot{\beta}}$ are a fuzzy combination of behavioural control signals u_{Ov} and $u_{O\dot{\beta}}$ for the OA behaviour, presented in Fig. 7(c), and control signals u_{Gv} and $u_{G\dot{\beta}}$ for the GS behaviour, shown in Fig. 7(d).

At the beginning of the movement, when the goal is far and obstacles are close to the WMRF, the main role in the trajectory generator control signals has the OA task, but at the end of the numerical test dominant are control signals obtained by the behavioural control system for the GS task.

Values of the actor's (W_{GA1}) and critic's (W_{GC1}) NN weights of ADHDP structure, that generates the behavioural control signal u_{GAv}, are shown in Figure 11(a) and (b). Weights of NNs are bounded and converge to the fixed values.

It was assumed, that the WMRF reached the goal if, in the end of the movement, the distance $d_G < 0.15$[m]. Using the trajectory generator with the cooperative connection approach it was not always possible to reach the goal with assumed accuracy. In the numerical test 2 the hierarchical control system with constant part of individual behaviours in the GS and OA tasks (a_B=const.) was not able to generate the successful trajectory of the WMRF, because of small distance of obstacles and the goal. In the other numerical tests accuracy of the trajectory generator with the cooperative connection approach was worse ($l_{G\{n\}} = 0.143$ [m] for the numerical test 1) than that of the trajectory generator with the fuzzy logic system ($l_{G\{k\}} = 0.013$ [m] for the numerical test 4).

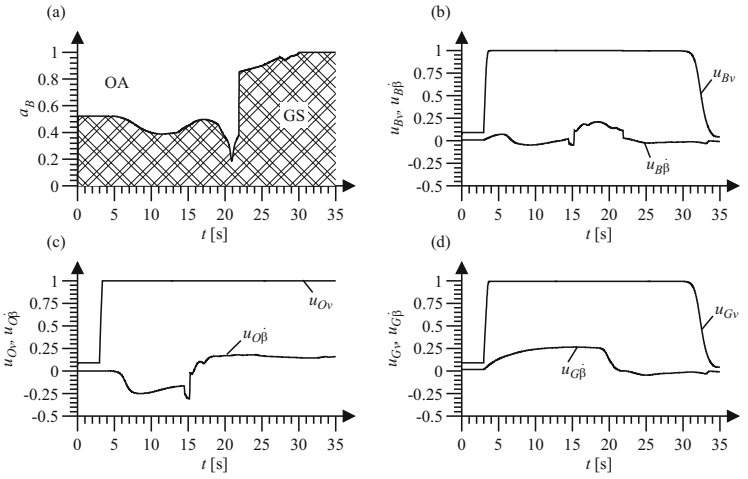


Fig. 7. (a) Signal a_B, (b) overall control signals u_{Bv} and $u_{B\dot{\beta}}$, (c) control signals u_{Ov} and $u_{O\dot{\beta}}$ for the OA behaviour, (d) control signals u_{Gv} and $u_{G\dot{\beta}}$ for the GS behaviour

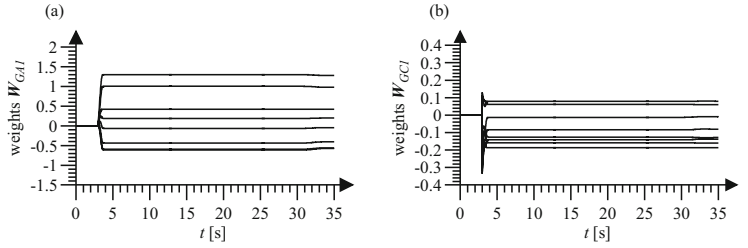


Fig. 8. (a) Weights of actor W_{GA1}, (b) weights of critic W_{GC1}

5 Summary

The proposed hierarchical control system, with NDP structures in ADHDP configuration in the trajectory generator and DHP algorithms in the tracking control system, generates and realises the collision free trajectory of WMRF in the unknown 2D environment with static obstacles. The trajectory generator consists of FL controller and two behavioural control systems for the OA and the GS behaviours. Each of the behavioural control algorithms consist of ACD algorithms and the proportional regulator, what is a new and innovative approach that prevents from the time consuming trial and error learning. The FL system generates control signal used to soft switching of the behavioural control signals. In the second versions of the trajectory generator, with the cooperative connection of behavioural control signals, a_B is constant. This second approach does not ensure successful completion of the complex tasks, when the goal is localised near obstacles. Trajectories generated using fuzzy logic system in the trajectory

generator provided, that the point M of the WMRF reached the goal in all tests. The shape of the formation was assured by the virtual structure control algorithm. The tracking control systems, individual for all agents, as the lowest layer of the hierarchical control system, realise trajectories generated by the trajectory generator. The projected hierarchical control system with the sensor-based navigator works on-line and does not require the preliminary learning of NNs.

References

1. Arkin, R.C.: Behavior-Based Robotics. MIT Press, Cambridge (1998)
2. Burghardt, A.: Proposal for a Rapid Prototyping Environment for Algorithms Intended for Autonomous Mobile Robot Control. Mechanics and Mechanical Engineering 12, 5–16 (2008)
3. Egerstedt, M., Hu, X.: Formation Constrained Multi-Agent Control. IEEE Transactions on Robotics and Automation 17, 947–951 (2001)
4. Giergiel, J., Hendzel, Z., Zylski, W.: Modeling and Control of Wheeled Mobile Robots. PWN, Warsaw (2002) (in polish)
5. Hendzel, Z., Burghardt, A., Szuster, M.: Artificial Intelligence Methods in Reactive Navigation of Mobile Robots Formation. In: 4th International Conference on Neural Computation Theory and Applications, pp. 466–473. SciTePress, Barcelona (2012)
6. Hendzel, Z., Szuster, M.: Discrete Action Dependant Heuristic Dynamic Programming in Wheeled Mobile Robot Control. Solid State Phenomena 164, 419–424 (2010)
7. Hendzel, Z., Szuster, M.: Discrete Model-Based Adaptive Critic Designs in Wheeled Mobile Robot Control. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 264–271. Springer, Heidelberg (2010)
8. Hendzel, Z., Szuster, M.: Neural Dynamic Programming in Reactive Navigation of Wheeled Mobile Robot. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 450–457. Springer, Heidelberg (2012)
9. Maaref, H., Barret, C.: Sensor-based Navigation of a Mobile Robot in an Indoor Environment. Robotics and Autonomous Systems 38, 1–18 (2002)
10. Millan, J.: Reinforcement Learning of Goal-Directed Obstacle-Avoiding Reaction Strategies in an Autonomous Mobile Robot. Robotics and Autonomous Systems 15, 275–299 (1995)
11. Orgen, P., Leonard, N.: Obstacle Avoidance in Formation. In: IEEE International Conference on Robotics and Automation, pp. 2492–2497. IEEE Press, Taipei (2003)
12. Powell, W.B.: Approximate Dynamic Programming: Solving the Curses of Dimensionality. Willey-Interscience, Princeton (2007)
13. Prokhorov, D., Wunch, D.: Adaptive Critic Designs. IEEE Transactions on Neural Networks 8, 997–1007 (1997)
14. Si, J., Barto, A.G., Powell, W.B., Wunsch, D.: Handbook of Learning and Approximate Dynamic Programming. IEEE Press, Wiley-Interscience (2004)
15. Tanner, H., Pappas, G., Kumar, V.: Leader to Formation Stability. IEEE Transactions on Robotics and Automation 20, 443–445 (2004)

State-Space Reduction through Preference Modeling*

Radosław Klimek, Igor Wojnicki, and Sebastian Ernst

AGH University of Science and Technology
Department of Applied Computer Science
Al. Mickiewicza 30, 30-059 Kraków, Poland
{rklimek,wojnicki,ernst}@agh.edu.pl

Abstract. Automated planning for numerous co-existing agents, with uncertainty caused by various levels of their predictability, observability and autonomy, is a complex task. One of the most significant issues is related to explosion of the state space. This paper presents a formal framework which can be used to model such systems and proposes the use of formally-modeled agents' preferences as a way of reducing the number of states. A detailed description of preference modeling is provided, and the approach is evaluated by examples.

1 Introduction

Automated planning is an active research field, with applications ranging from motion planning [1], through resource allocation and scheduling, to coordination of autonomous agents [2]. This paper presents a formal framework which can be used to model sophisticated systems, featuring heterogenous entities (agents), characterized by varying levels of autonomy, predictability and observability, and proposes the use of preferences to reduce the size of the state space.

The paper is organized as follows. Section 2 provides a brief introduction to automated planning and the state-space representation of planning problems. Section 3 presents the framework, indicating certain and uncertain knowledge elements, and provides intuition how this model can be mapped to real-world cases. Section 4 introduces a formal tool which can be used to model preferences (or agent constraints) using temporal logic, and Section 5 provides an example how formally represented preferences can be used to reduce the state space.

2 Motivation and State-of-the-Art

The most general, intuitive definition of planning is that it is the *reasoning part of acting* [3]. Planning can be performed by people, either implicitly or explicitly. Automated planning is a branch of AI which is concerned with computation and execution of plans by machines.

* This work is supported by the Polish National Science Centre (NCN) grant 2011/01/D/ST6/06146.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 363–374, 2013.

A common conceptual model for planning is a *state-transition system*, also called *discrete-event system*. It can be defined as a 4-tuple $\Sigma = (S, A, E, \gamma)$ [3,4], where:

- $S = \{s_1, s_2, \ldots\}$ is a set of *states*,
- $A = \{a_1, a_2, \ldots\}$ is a set of *actions*,
- $E = \{e_1, e_2, \ldots\}$ is a set of *events*,
- $\gamma : S \times A \times E \to 2^S$ is a state-transition function.

This definition is often used together with a graph model, where states $s \in S$ are nodes, and state transitions (given as pairs $(a, e), a \in A, e \in E$) are directed edges. The semantic difference between actions and events is that actions are *applicable* to states by the plan executor (if $\gamma(s, a) \neq \emptyset$), while events are *contingent* – they might occur due to the system's characteristics, and every event e which takes the system from state s to state s' must have a corresponding transition function $\gamma(s, e) = \{s'\}$. In a typical instance of a planning problem, the system is in some initial state $s_I \in S$, and the goal is to take it to one of the goal states $s_G \in \{s_{G1}, s_{G2}, \ldots\} \subset S$.

Based on that definition, derivation of a plan consists in finding a sequence of state transitions which take the system form the initial state to one of the goal states. Numerous methods can be used here – either "blind" (uninformed) ones, which do not take the characteristics of the system into account [4, sec. 3.4] or informed (heuristic) methods, which are especially useful for large state spaces [5]. Determination of heuristics for *domain-independent planning* is a problem which has recently received a lot of attention [6,7,8,9].

The size of the state space and the complexity of the planning process can grow rapidly, due to several reasons:

1. **Partially predictable action (and event) results.** If it is unclear which state the system will be in after an action is taken or an event occurs, the planner will have to consider all possible outcomes, which may result of a combinatorial explosion of the search tree.
2. **Partially determinable world state.** If the world state cannot be fully observed, the planner needs to assume all possible states which match the observations.
3. **Multiple agents.** The size of the state space grows exponentially with the number of entites (agents), as the local state of an agent can be combined with almost all states of every other agent. Planing for such a case (multi-body) is not different than for a single agent though[4].
4. **Multi-variant plans.** Sometimes, it may be preferable to provide multiple possible actions to the agent and let it choose the most favorable option. This is true especially in situations, where the planner's observations of the system state are less detailed than (local) observations made by the agents.

In needs to be noted, that the state space is not a Cartesian product of all possible states. Some states can be unreachable due to world constraints (obstacles) or agent constraints restricting its behavior. Defining these constrains decreases planning process complexity.

3 Considered World

This section presents the general model of a world of coexisting entities and their supervisor. The entities differ with regard to the following parameters:

- *controllability* – the extent to which the supervisor may influence the entities; inversely proportional to the entities' *autonomy*,
- *predictability* – certainty that the entity will act according to the prediction, either due to its intents or its abilities to fulfill the orders,
- *observability* – the ability of the supervisor to observe the actions of an entity, whether autonomous or performed to fulfill an order.

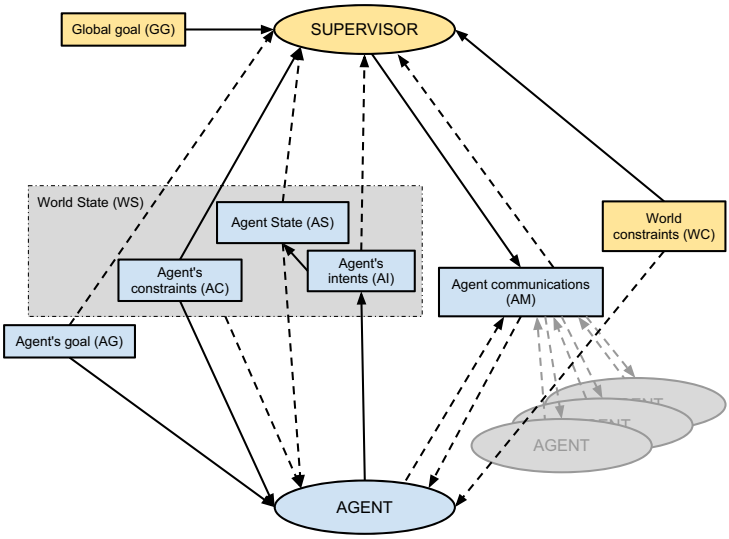


Fig. 1. Knowledge in the world model. Solid lines indicate certain knowledge, while dashed lines indicate knowledge that is subject to an arbitrary level of uncertainty.

An overview of the knowledge existing in the world and its availability to particular entities has been presented in Figure 1. The knowledge elements are as follows:

Global Goal (GG). The desired state of the world, known to the *Supervisor*, irrelevant to *Agents*.

World State (WS). State of other agents (AC, AS, AI) as perceived or inferred by the *Agent* under consideration.

Agent's State (AS). Current state of an individual agent, might be known to the *Agent* or *Supervisor*. It is an effect of the agent's actions, which result from the Agent's Intents (AI).

Agent Communications (AM). May occur:

- between the *Supervisor* and the *Agent*, typically including *orders* (downstream) and *order fulfilment acknowledgements* (upstream),
- between individual agents, typically including negotiations and knowledge broadcast.

Agent's Constraints (AC). Limitations of an *Agent* with regard to actions it can perform; known to the *Agent* and the *Supervisor*.

Agent's Goal (AG). Goal of an individual *Agent*. Known to the *Agent*, might be known to the *Supervisor*.

Agent's Intents (AI). Generated by the *Agent* based on its goal (AG), received orders (subset of AM) and state of other agents (WS).

World Constraints (WC). Known to the *Supervisor*, may be known (e.g. observed) by the *Agent*.

Orders, received within AM, are usually plans, often multi-variant ones, i.e. consisting of more than one path leading to the destination. Since the proposed model is generic, potential applications regard many different domains, both abstract and very particular.

For more abstract solutions, multi-agent systems are targeted. These include problem decomposition and agent scheduling. The issue of problem decomposition regards subdividing the problem into manageable sub-problems and ascribing them to given agents. Depending on the nature of the problem, it may be required to perform subdivision in a planned way. Synthesizing multiple plans and choosing the optimal or suboptimal one poses a challenge. Similarly, while managing multiple agents towards reaching a global goal, there is a need to plan their actions. In such case, the aforementioned issues of controllability, observability and predictability come to play. Furthermore, in many cases, such planning should also be multi-variant, especially if communication with some agents is not reliable.

There are many similarities between pure software multi-agent systems and real-world applications. One of the cases to be considered here is fleet management (in terms of a supply chain, consisting of trucks, trains, cargo ships, etc.). While there might be a well defined goal in existence, which formulates optimization criteria from the perspective of managing shipments, goals of individual agents can fluctuate in time and space. Such fluctuations might be caused by a wide range of factors, including technical issues with vessels, natural disasters, communication interruptions, or human factor issues.

Search and rescue operations can also be considered here. They form a complex system which includes information about the environment under consideration, rescue team members, supporting hardware (such as autonomous search and mapping robots), domain experts, other personnel (e.g. volunteers), commanding centers, etc. Since such operations are carried out under extreme variability of factors regarding the system components, it is not safe to assemble a single plan. Instead, multiple plans for such cooperative activities should be established to compensate for lack of controllability, observability or predictability.

Yet another application are team sports, both these performed by humans (e.g. basketball) or robots (such as RoboCup soccer). The ultimate goal here is

to win. There are multiple agents (the players) performing actions and carrying out orders and suggestions and the couch, or programmer, as the supervisor. Depending whether it is a simulated game, a RoboCup competition or a real one, the proposed World still holds and expresses the system.

4 Preference Modeling

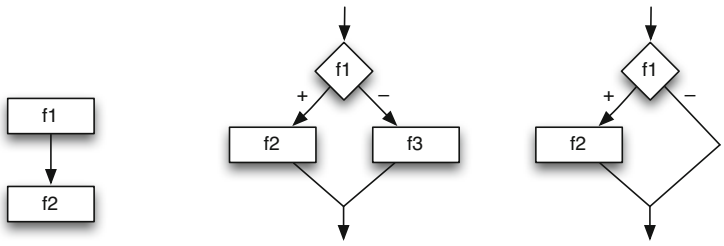
Preference modeling and choice prediction is crucial in the world of agents. *Preference* is an act of selecting something over others possibilities of choice. It is equivalent to the principle of giving advantages to some options over others. It also enables customization of the agents' behavior and preferences could be used to choose from a multi-variant plan. Decisions of agents require a mechanism for representing and reasoning about the possible consequences of their choices. Logical models seem especially appropriate for representing preference models and their exploration. Flexible models should provide the potential for interpreting preference choices in a way that does not depend on the underlying utility model.

Preference modeling and preference models need formalization and are discussed in some works, e.g. work by Öztürk et al. [10]. The preference models might be constructed using fuzzy sets, classical logic and many-valued logics. Classical logic, particularly rule-based systems, are especially popular, c.f. [11]. Non-classical logics, especially temporal logic, are less popular here. However, let us note that temporal logic is a well established formalism for describing system reactiveness. On the other hand, many applications are characterized by reactivity and flexibility in adapting to changes on the user side. These changes may result from recognized and predefined preferences which are applied in practice. The variability and change in valuation of logical statements over time flows are difficult to describe in classical logic. This is a reason important enough to include temporal logic in the considerations for preference models. Temporal logic creates new possibilities for analysis of preferences by going beyond the static world of classical logic. It also allows to illustrate the dynamic aspect of preferences which describe situations of preference valuations that vary over flows of time. The temporal approach seems to be underestimated in preference modeling. After building a preference model in temporal logic, one can analyze it using a deductive approach. The goal is searching for contradictions, if any, in a model. It is also possible to inference about the correctness of preference objectives.

Temporal Logic TL is a formalism, e.g. [12], which has strong application for specification and verification of models. It exists in many varieties; however, considerations in this paper are limited to the *Linear Temporal Logic* LTL, i.e. logic for which the time structure is considered linear. Considerations in this paper are limited to the *smallest temporal logic*, e.g. [13]. This logic is considered a classical propositional calculus' extension of axiom $\Box(P \Rightarrow Q) \Rightarrow (\Box P \Rightarrow \Box Q)$ and inference rule $\frac{\vdash P}{\vdash \Box P}$. The following formulas may be considered as significant examples of the minimal temporal logic: $LowVoltageDetection \Rightarrow \Diamond Alarm$,

$\Box(Trigger \Rightarrow \Diamond Action)$, $\Diamond Live$, $\Box\neg(BadEvent)$ or $\Box\neg(Event1 \wedge (Event2 \vee Event3))$, etc.

Semantic tableaux is a decision procedure for checking formula satisfiability. The method is well known in classical logic, but it can also be applied in modal and temporal logics [14]. The method is based on formula decompositions. At each step of a well-defined procedure, formulas are decomposed and have fewer components since logical connectives are removed. Finding a contradiction in all branches of the decomposition tree means there are no valuations that satisfy a formula placed in the root. When all branches of the tree have contradictions, it means that the inference tree is *closed*. If the negation of the initial formula is placed in the root, this leads to the statement that the initial formula is true. This method has some advantages over the traditional axiomatic approach. In the classical reasoning approach, starting from axioms, longer and more complicated formulas are generated and derived. Formulas become longer and longer step by step, and only one of them will lead to the verified formula. The method of semantic tableaux is characterized by a reversed strategy. The inference structure is represented by a tree and not by a sequence of formulas. Expansion of any tree branch may be halted after finding a contradiction. In addition, the method provides, through so-called *open* branches of the semantic tree, information about the source of an error, if one is found; that is another and very important advantage of the method. The work [15] shows an example of the truth tree of the semantic tableaux method for minimal temporal logic.



(a) Sequence (b) Branch (c) SimpleBranch

Fig. 2. Patterns for preferences

The proposed preference model is based on preference patterns. A *pattern* is a predefined solution for a special context of preference issues. Preference patterns are shown in Fig. 2 and are introduced in the work [15]. Every preference model, perhaps elicited during the requirements engineering phase, consists only of these patterns. Preference patterns can be nested and form complex models. Preferences are in the form of logical rules expressed in temporal logic. Suppose that well-formed and syntactically correct temporal logic formulas are already defined, c.f. [12].

Patterns constitute a kind of primitives. They are indicated as $pat()$, where pat is a name of a given pattern, and their parameters, if any, are included in parentheses. The following three patterns are considered: *Branch*, *SimpleBranch* and *Sequence*. They constitute a kind of illustration of the if-then scheme. Pattern nesting enables projection of a multi-stage decision-making. Say, a *basic set of patterns* Σ is a set of temporal logic formulas describing properties of a pattern. Thus, a set of three patterns, i.e. $\Sigma = \{Branch, SimpleBranch, Sequence\}$, is considered. Now, let us define *temporal properties* $\Pi(\Sigma)$ over predefined patterns Σ. Hence, set $Branch(f_1, f_2, f_3) = \{c(f_1) \Rightarrow \Diamond f_2 \wedge \neg\Diamond f_3, \neg c(f_1) \Rightarrow \neg\Diamond f_2 \wedge \Diamond f_3, \Box\neg(f_1 \wedge (f_2 \vee f_3))\}$ describes properties of the Branch pattern and set $SimpleBranch(f_1, f_2) = \{c(f_1) \Rightarrow \Diamond f_2, \neg c(f_1) \Rightarrow \neg\Diamond f_2, \Box\neg(f_1 \wedge f_2)\}$ the SimpleBranch pattern. Set $Sequence(f_1, f_2) = \{f_1 \Rightarrow \Diamond f_2, \Box\neg(f_1 \wedge f_2)\}$ defines the Sequence pattern. Formulas that constitute set for a patterns describe both liveness and safety property of a pattern. Formulas f_1, f_2 etc. are atomic formulas for a pattern. They are a kind of formal arguments for a pattern. $\Diamond f$ means that sometime activity f is completed, i.e. when the token left the activity, i.e. when the falling edge (or the negative edge) of the activity is transited. $c(f)$ means that the logical condition associated with activity f has been evaluated and is satisfied. This logical condition is satisfied when the falling edge is transited. There is a standard taxonomy of system properties, i.e. liveness and safety, which are adapted here to the field of preferences:

- *liveness* means that some preferences might be achieved, if desired, in the whole preference model, e.g. $p_1 \Rightarrow \Diamond p_5$ or $\Diamond p_4$;
- *safety* means that some preferences, if any, perhaps a logical combination of a subset of the entire preference world, are avoided; e.g. $\Box\neg(p_2 \wedge \neg p_6 \wedge p_{10})$.

where $\{p_1, ..., p_{10}\}$ is a whole identified preference world.

The entire preference model can be written in the form of logical expressions in order to write preferences in a concise and literal notation. The *logical expression* W_L is a structure created using the following rules:

- every elementary set $pat(a_i)$, where $i > 0$ and every a_i is an atomic formula, is a logical expression,
- every $pat(A_i)$, where $i > 0$ and every A_i is either (a) an atomic formula, or (b) a logical expression $pat()$, is also a logical expression.

Any logical expression may represent an arbitrary structure of patterns and an example of this are the following expressions: $Branch(a, SimpleBranch(f, g), c)$ and $Sequence(Branch(a, b, c), SimpleBranch(d, e))$. In the first case, the combination (and nesting) of two branched patterns is considered, a and f are conditions in this expression. In the second case, the sequence of two branched patterns is considered. Individual preferences expressed as a logical expression may belong to a set of preferences R, i.e. $R = \{r_1, r_2, ...r_n\}$, where every r_i is a preference which is expressed as a single logical expression.

When building a logical model for preferences, two important aspects of a logical system can be analyzed: (1) semantic contradiction of a model, or (2)

correctness of the model due to some properties. Formal verification of properties for a preference model leads to the analysis of the formula $s_1 \wedge \ldots \wedge s_n \Rightarrow Q$, where Q is a desired property for the preference model $\{r_1, r_2, ...r_m\}$.

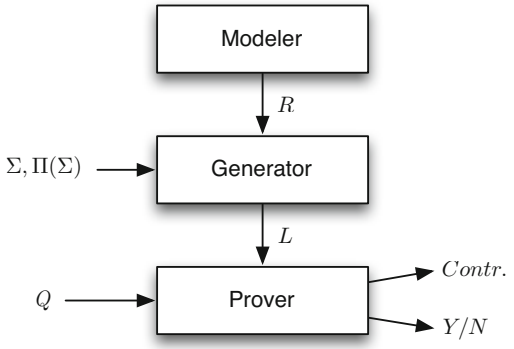


Fig. 3. Deduction system for preference models

The architecture of a system for automatic inference on preference models is proposed in Fig. 3. The Modeler module allows to prepare a preference model using preference patterns shown in Fig. 2. The output of the Modeler are preference models R expressed as logical expressions. The next module is the Generator module. The inputs of the Generator are logical expressions R and a predefined set of basic preference patterns Σ, together with their predefined temporal properties $\Pi(\Sigma)$. The output is a logical specification L understood as a set of temporal logic formulas. The sketch of the generating algorithm is the following:

1. At the beginning, the logical specification is empty, i.e. $L = \emptyset$;
2. The most nested pattern or patterns are processed first, then, less nested patterns are processed one by one, i.e. patterns that are located more towards the outside;
3. If the currently analyzed pattern consists only of atomic formulas, the logical specification is extended, by summing sets, by formulas linked to the type of the analyzed pattern $pat()$, i.e. $L = L \cup pat()$;
4. If any argument is a pattern itself, then the logical disjunction of all its arguments, including nested arguments, is substituted in place of the pattern.

The above algorithm refers to similar ideas in work [16]. Let us supplement the algorithm by some examples. The example for the step 3: $Seq(p, q)$, gives $L = \{p \Rightarrow \Diamond q, \Box\neg(p \wedge q)\}$ and $Branch(a, b, c)$ gives $L = \{c(a) \Rightarrow \Diamond b \wedge \neg\Diamond c, \neg c(a) \Rightarrow \neg\Diamond b \wedge \Diamond c, \Box\neg(a \wedge (b \vee c))\}$. The example for the step 4: $Sequence(Branch(a, b, c), d)$ leads to $L = \{c(a) \Rightarrow \Diamond b \wedge \neg\Diamond c, \neg c(a) \Rightarrow \neg\Diamond b \wedge \Diamond c, \Box\neg(a \wedge (b \vee c)), (a \vee b \vee c) \Rightarrow \Diamond d, \Box\neg((a \vee b \vee c) \wedge d)\}$.

The Prover module works using the semantic tableaux method described above. The inputs for the Prover include a logical specification $L = \{s_1, ..., s_n\}$ and a query Q which might be a simple temporal logic formula expressing the desired property for the preference model. (This formula can be introduced using a simple text editor.) The Prover provides two kinds of answers which originate from two following cases:

1. Semantic contradiction, i.e. analysis of the formula:

$$s_1 \wedge \ldots \wedge s_n \tag{1}$$

2. Correctness of the model due to some properties, i.e. the formal verification of the formula:

$$s_1 \wedge \ldots \wedge s_n \Rightarrow Q \tag{2}$$

In the case of contradiction, formula 1 is placed in the root of the reference tree and the information about the semantic contradiction is produced. In the case of correctness, the negation of the formula 2 is placed in the root of the reference tree and the *Yes/No* output is produced.

Let us consider a simple example for the house domain. Preferences can be aggregated over groups of two objectives. The first one is the monitoring and alarm system. When danger is detected, a security group or the fire brigade is called depending on the type of danger; however, in both cases the police is called. After modeling preferences, one can obtain the following logical expression:

$$\begin{aligned} r_1 = Sequence(&Danger, Sequence(Branch(\\ &Intrusion, SecurityGroup, SimpleBranch(\\ &Fire(FireBrigade)))), Police) \end{aligned} \tag{3}$$

The second objective is related to the method of house heating. Let us consider two different intervals of hours. Nights have lower range of temperatures in contrast to a higher range of temperatures for mornings and afternoons. Finally, one can obtain:

$$\begin{aligned} r_2 = Branch(&Weekend, Branch(Night1, Lower, \\ &Higher), Branch(Night2, Lower, Higher)) \end{aligned} \tag{4}$$

Thus, $P = \{Danger, Intrusion, ..., Weekend, Night1, ...\}$ are atomic preferences which were identified when preparing preference model. Next, $R = \{r_1, r_2\}$ are preferences considered as a set of logical expressions. Finally, the logical specification L is generated using the algorithms described above with two inputs, i.e. preferences R and the set of predefined patterns Σ together with related temporal properties $\Pi(\Sigma)$.

5 Examples of State-Space Reduction Cases

Let us consider a fully observable, predictable, and controllable world with given constraints, the Grid World. Knowledge regarding it is considered as follows, particular world components given in parentheses refer to Fig. 1.

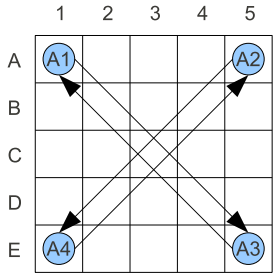


Fig. 4. Initial state of the proposed Grid World

The world is two-dimensional, based on a five by five grid (WC). The agents, denoted as A_n can move from one location to another orthogonally only, in accordance with the von Neumann neighborhood: north, south, east and west (WC). No more than one agent is allowed at a single location. There are four agents initially located in the corners (AS) (see Fig. 4). There is no inter-agent communication (AM). Each agent's goal it to get to the diagonally opposite corner (AG); A_1 to $(E, 5)$, A_2 to $(E, 1)$, A_3 to $(A, 1)$, A_4 to $(A, 5)$. Global goal (GG) is to guide all the agents to their goals corners. Since the world is fully controllable, the agents' intents comply with orders given by the supervisor (AM).

Assuming the above, a state consists of information about agent locations, thus it is a tuple: $s = (x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$. Assuming that each agent can be at any location, having 5×5 grid, cardinality of S, number of states, is given as:

$$|S_1| = (5 * 5)^4 = 390625.$$

Applying the world constraints (no two agents sharing the same location) reduces number of states slightly, having a product of arithmetic progression:

$$|S_2| = (5 * 5) * (5 * 5 - 1) * (5 * 5 - 2) * (5 * 5 - 3) = 303600.$$

Let us now introduce preferences in order to demonstrate how they can reduce the state space. Some agents suffer from *demophobia*[1], i.e. avoid being in a crowd. The consequence of this is an appropriate model of preferences. However, let us consider the whole world of atomic preferences

$$P = \{Go(N), Go(E), Go(S), Go(W), Crowd(d), ...\} \tag{5}$$

where $Go(N)$ means the agent preference to go North (or East, South, West), etc. Function $Crowd(d)$ is a Boolean function that tests the presence of the crowd, where d means a direction North, East, South or West and express the desired direction for agent.

Let us express the following preference model for an agent. He examines all directions, and only as a last resort go to fourth direction. Suppose that d is the current direction which expresses the agent desire and its will.

[1] A fear of crowds, masses, or people.

$$r_1 = Branch(Crowd(d), Branch(Crowd(d+1), Branch(Crowd(d+2), Go(d+3), Go(d+2)), Go(d+1)), Go(d)) \quad (6)$$

After examining all the directions, we go in the last possible one.

If we take the preference described above into account, the state space will be reduced to at most $|S_3| = (5*5)*(5*5-3)*(5*5-6)*(5*5-9) = 167200$. Please note that the number above is the upper bound of the number of states, based on the assumption that the presence of an agent renders at most three of its neighboring cells unfavorable for other agents, in order to simplify calculations.

More detailed analysis of all possible combinations has revealed that the actual number of states fulfilling the above preference is 120576.[2] This indicates significant state reduction compared to $|S_2|$ ($\approx 60\%$) and $|S_1|$ ($\approx 69\%$). As the complexity and number of preferences is arbitrary, the deduction system (see Section 4) can use them to identify states which are unreachable and eliminate them, thus reducing the state space.

6 Conclusions and Future Work

Planning for multiple entities causes an explosion of the number of states in the state space. Variations of agent controllability, observability and predictability add to the problem by introducing belief states and states which would not otherwise be considered by the supervisor.

The number of states can be reduced by identifying unreachable states, which in turn can be eliminated from the state space. This can be achieved by defining and analyzing agent's preferences. As shown in Section 5, some preferences can influence the number of states in terms of their reduction.

Moreover, even if states cannot be eliminated from the state space, formally defined preferences can be useful if the agent receives multi-variant instructions and needs to decide which path to follow.

Other preferences may not eliminate the possible system states, but can influence the transitions between them. An intuitive example could be a preference of an agent not to turn left. While the number of states would not necessarily be reduced (as one left turn could be compensated by multiple right turns), they can be used by the planner to prune the search tree, thus reducing its branching factor.

Future work includes perfecting identification of unreachable states and improvement of the formalism used to model preferences, as well as research related to preference-driven search or planning algorithms and extraction of heuristics from preference sets.

References

1. Turek, W., Marcjan, R., Cetnarowicz, K.: Software agent systems for improving performance of multi-robot groups. Fundamenta Informaticae 112(1), 103–117 (2011)

[2] Value obtained by means of numerical analysis.

2. Cetnarowicz, K.: From algorithm to agent. In: Allen, G., Nabrzyski, J., Seidel, E., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2009, Part II. LNCS, vol. 5545, pp. 825–834. Springer, Heidelberg (2009)
3. Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory & Practice. Morgan Kaufmann Publishers Inc., San Francisco (2004)
4. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Pearson Education (2010)
5. Bonet, B.: Planning as heuristic search. Artificial Intelligence 129(1-2), 5–33 (2001)
6. Keyder, E., Geffner, H.: Trees of shortest paths vs. Steiner trees: Understanding and improving delete relaxation heuristics. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009), pp. 1734–1749 (2009)
7. Karpas, E., Domshlak, C.: Optimal Search with Inadmissible Heuristics. In: International Conference on Automated Planning and Scheduling, pp. 92–100 (2012)
8. Haslum, P.: hm (P)= h1 (Pm): Alternative characterisations of the generalisation from hmax to hm. In: Proc. ICAPS, vol. 1, pp. 354–357 (2009)
9. Helmert, M., Domshlak, C.: Landmarks, critical paths and abstractions: Whats the difference anyway. In: Proc. ICAPS, vol. 9 (2009)
10. Öztürk, M., Tsoukiàs, A., Vincke, P.: Preference modelling. In: Figueira, J., Greco, S., Ehrgott, M. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys, pp. 27–72. Springer, Boston (2005)
11. Fong, J., Indulska, J., Robinson, R.: A preference modelling approach to support intelligibility in pervasive applications. In: 8th IEEE Workshop on Context Modeling and Reasoning (CoMoRea 2011), Seattle, USA, March 21-25, pp. 409–414. IEEE (2011)
12. Wolter, F., Wooldridge, M.: Temporal and dynamic logic. Journal of Indian Council of Philosophical Research XXVII(1), 249–276 (2011)
13. Chellas, B.F.: Modal Logic. Cambridge University Press (1980)
14. d'Agostino, M., Gabbay, D.M., Hähnle, R., Posegga, J.: Handbook of Tableau Methods. Kluwer Academic Publishers (1999)
15. Klimek, R.: Temporal preference models and their deduction-based analysis for pervasive applications. In: Benavente-Peces, C., Filipe, J. (eds.) Proceedings of 3rd International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS 2013), Barcelona, Spain, February 19-21, pp. 131–134. SciTePress (2013)
16. Klimek, R.: Proposal to improve the requirements process through formal verification using deductive approach. In: Filipe, J., Maciaszek, L. (eds.) Proceedings of 7th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2012), Wrocław, Poland, June 29-30, pp. 105–114. SciTePress (2012)

Multi-Agent Temporary Logic $TS4^{U}_{K_n}$ Based at Non-linear Time and Imitating Uncertainty via Agents' Interaction

David McLean[1] and Vladimir Rybakov[1,2]

[1] School of Computing, Mathematics and DT,
Manchester Metropolitan University,
John Dalton Building, Chester Street, Manchester M1 5GD, U.K.
[2] Institute of Mathematics, Siberian Federal University,
79 Svobodny Prospect, Krasnoyarsk, 660041, Russia
{D.McLean,V.Rybakov}@mmu.ac.uk

Abstract. This paper considers AI problems concerning reasoning in multi-agent environment. We introduce and study multi-agents' non-linear temporal logic $\mathbf{TS4^{U}_{K_n}}$ based on arbitrary (in particular, non-linear, finite or infinite) frames with reflexive and transitive accessibility relations, and individual symmetric accessibility relations R_i for agents. Main accent of our paper is modeling of logical uncertainty for statements via interaction of agents (passing knowledge). Conception of interacting agents is implemented via arbitrary finite paths of transitions by agents accessibility relations. We address problems decidability and satisfiability for $\mathbf{TS4^{U}_{K_n}}$. It is proved that $\mathbf{TS4^{U}_{K_n}}$ is decidable (and, in particular, the satisfiability problem for it is also decidable). We suggest an algorithm for checking satisfiability based on computation possibility of refutation special inference rues in finite models of effectively bounded size.

Keywords: multi-agents' logic, interacting agents, temporal logic, non-linear temporal logic.

1 Introduction

Basically a multi-agent system (MAS) is a system composed of multiple interacting intelligent agents within certain environment. Study of MASs (with autonomous or interacting, say competitive) agents is an active area in modern AI. Technique and research outputs are various, diverse and work well in many contemporary areas (though, it seems, most popular area is applications in IT, – cf. Nguyen et al [17–19], Arisha et al [1], Avouris [2], Hendler [11]). Area of modeling reasoning (initially, – an individual human reasoning) is an old branch of AI, which now includes technique for modeling multi-agents reasoning. These techniques use a logical language for reasoning about agents' knowledge and properties (e.g. various technique of mathematical (symbolic) logic is widely used (cf. [10, 12, 13])); in particular, multi-agent modal logics were implemented.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 375–384, 2013.

Historically, multi-agent epistemic logics have found various applications in fields ranging from AI domains such as robotics, planning, and motivation analysis in natural language, to negotiation and game theory in economics, to distributed systems analysis and protocol authentication in computer security. Ability of intelligent agents to reason about knowledge is an essential feature of those applications. For instance, technique of non-classical logic (as modal, temporal or epistemic) gives inference capabilities to deduce implicit knowledge from the explicitly represented facts. This approach allows to describe the properties (specifications) with explicit, mathematically preciseness, which simplifies identification. Historically, usage of logical language in knowledge representation is known since reasonably long ago (cf. Brachman and Schmolze (1985, [7]), Moses and Shoham (1993, [14]), Nebel (1990, [15]). It also had some applications in industry Rychtycki (1996, [30]).

To represent knowledge and to specify it, the question what is a shared knowledge and what is a common knowledge for all agents has been risen. It seems, first ideas concerning this specification appeared in Barwise (1988, [8]), Niegerand and Tuttle (1993, [16]), Dvorek and Moses (1990, [9]). Modern approach to common knowledge logics was suggested in the book Fagin R., Halpern J., Moses Y., Vardi M. (1995, [10]). This book contains a series of theorems on completeness for various common knowledge logics w.r.t. possible worlds models.

In study of multi-agents' reasoning, an essential point is how to represent interaction of agents, exchange of information (cf. e.g., Sakama et al [20]). Study of multi-modal agents logics and temporal agents-logics, representing these features, were undertaken in a series of works by V.Rybakov. A kernel part in these works was a representation the case when the logics describe interacting agents. In Rybakov, 2009, [26] some technique to handle interactions was found, and, as a consequence, it was proved that the multi-agent Linear Temporal Logic (with UNTIL and NEXT and with interacting agents, or dually, common knowledge) is decidable, that the satisfiability problem for this logic is also decidable, and some algorithms solving the problem were found (cf. also Rybakov [25]). Besides, research of just multi-agent logics (as modal and temporal) with aim to find solution of satisfiability problem (and decidability corresponding logics) was earlier undertaken in Rybakov [27, 28], Babenyshev and Rybakov [3–6].

This paper studies the multi-agents' non-linear temporal logic $\mathbf{TS4}^{\mathbf{U}}_{\mathbf{K_n}}$ based on arbitrary (in particular, non-linear, finite or infinite) frames with reflexive and transitive accessibility relations, and individual symmetric accessibility relations R_i for agents. The impellent aim of our paper is how to represent logical uncertainty in multi-agent reasoning. We suggest to model logical uncertainty for statements via interaction of agents (passing knowledge).

Conception of interacting agents is implemented via arbitrary finite paths of transitions by agents accessibility relations. This approach uses technique developed by V.Rybakov in the mentioned above papers, it essentially uses [26], and the current paper extends results from [29] in order to handle logical uncertainty via interaction of agents. Main computational problems we dealing with are problems of decidability and satisfiability for $\mathbf{TS4}^{\mathbf{U}}_{\mathbf{K_n}}$. We show that $\mathbf{TS4}^{\mathbf{U}}_{\mathbf{K_n}}$

is decidable (in particular, the satisfiability problem for it is also decidable). An algorithm for checking satisfiability based on computation truth values of special inference rues in finite models of effectively bounded size is constructed. Paper contains some preliminary information for understanding obtained results.

2 Basic Notation, Definitions, Known Facts

We start from recalling notation and known facts concerning modal, multi-modal and temporal logics (so, some familiarity with these areas is assumed, though we give below the definitions to follow the paper). First, we the need special Kripke/Hintikka-like frames **F** defined as follows: $\mathbf{F} := \langle W, R, R_1, \ldots, R_n \rangle$, where W is a set of states (symbols of states, modeling web sites), R is a binary relation on W (modeling, for example, web connections, or runs of computations): aRb means that there is a connection from state a to state b (e.g. by clicking link buttons, some amount of steps in a computational procedure, etc.).

Relation R is assumed to be reflexive and transitive (which corresponds well to understanding time in a run of a computation, and models transitions in runs of computations, passing via web connections, etc). Thus, the following holds: $\forall a \in W,\ aRa; \quad \forall a, b, c \in W,\ aRb\ \&\ bRc\ \Rightarrow\ aRc$. States (worlds) in **F** – symbols from W – form with respect to R clusters. A cluster $C(a)$ generated by $a \in W$ is the set $\{b \mid b \in W, aRb\ \&\ bRa\ \}$.

Any relation R_i (agent i accessibility relation) is reflexive, transitive and symmetric relation (i.e. $aR_ib \Rightarrow bR_ia$) on $C(a)$ for any $a \in W$. An interpretation of such approach to model agents' relation via internet connections is as follows: being logged at web-site a, i-agent may access by R_i some web sites from $C(a)$ (in accordance with possession of access rules/passwords) - and switch between sites in its disposal freely, back and forth. Yet i cannot jump to another sites outside $C(a)$ without permitting (convoy) from administrator.

For computational runs the interpretation is similar: there are several computational threads imitated as relations R_i – any thread is a computational agent, relation R holds a cluster of local computations around an time tick. We would like to built a framework for description multi agent reasoning based at such sematic approach.

Our approach is based at a hybrid of a non-linear temporal logic and some knowledge multi-agent logic. Language of our logic consists of standard language of Boolean logic extended with temporal and agent knowledge operations. So, it contains potentially infinite set of propositional letters P; the logical operations include usual Boolean logical operations and usual unary agent knowledge operations K_i, $1 \leq i \leq m$; also it contains the operation for knowledge via agents' interaction **KnI** (this operation may be expressed as a dual counterpart of the common knowledge operation introduced, e.g. in Fagin et al [10]), and the unary logical operation U with meaning 'uncertain'.

Temporal unary operations are $\Diamond^+$ (with meaning 'possible in future' by a sequence of computational steps) and $\Diamond^-$ (with meaning *possible, so to say*

in past, – by a sequence of backtracks). The formation rules for formulas are standard: any propositional letter is a formula,

(i) if α and β are formulas, then $\alpha \wedge \beta$ is a formula;

(ii) if α and β are formulas, then $\alpha \vee \beta$ is a formula;

(iii) if α and β are formulas, then $\alpha \to \beta$ is a formula;

(iv) if α is a formula, then $\Diamond^{+}\alpha$ is a formula;

(v) if α is a formula, then $\Diamond^{-}\alpha$ is a formula;

(vi) if α is a formula, then for any i $K_i\alpha$ is a formula;

(vii) if α is a formula, then $\mathbf{KnI}\alpha$ is a formula;

(vii) if α is a formula, then $\mathrm{U}\alpha$ is a formula.

Intuitive meaning of this operations is as follows.

$K_i\varphi$ can be read: agent i knows φ in the current state;

$\Diamond^{+}\varphi$ says that there is a state (web site) b accessible from the current state a by a sequence of links, were the statement (formula) φ is true at b. So to say, there is a state, accessible in future, where φ is true.

$\Diamond^{-}\varphi$ means that there is a state b accessible from the current state a by a sequence of backtracks, were the statement (formula) φ is true at b.

$\mathbf{KnI}\varphi$ means: in the current state, the statement φ *may be known by interaction between agents.*

$\mathrm{U}\varphi$ has meaning the statement φ is uncertain (has uncertain truth value).

Next step of our construction are rules to compute truth values of formulas at states of arbitrary frames (where some truth valuation for formulas' letters is given; such frames with given valuations we will call models). So, given a frame $\mathbf{F} := \langle W, R, R_1, \ldots, R_n\rangle$, and a set of propositional letters P, a valuation V of P in $\mathbf{F}$ is a mapping of P into the set of all subsets of the set W, in symbols, $\forall p \in P, V(p) \subseteq W$. If, for an element $a \in W$, $a \in V(p)$ we say *the fact* p *is true in the state* a. In the notation below $(\mathbf{F}, a) \Vdash_V \varphi$ is meant to say the formula φ in true at the state a in the model $\mathbf{F}$ w.r.t. the valuation V. The rules for computation of truth values of formulas are as follows:

$$\forall p \in P,\ \forall a \in W\ \ (\mathbf{F}, a) \Vdash_V p \iff a \in V(p);$$

$$(\mathbf{F}, a) \Vdash_V \varphi \wedge \psi \iff [(\mathbf{F}, a) \Vdash_V \varphi \text{ and } (\mathbf{F}, a) \Vdash_V \psi];$$

$$(\mathbf{F}, a) \Vdash_V \varphi \vee \psi \iff [(\mathbf{F}, a) \Vdash_V \varphi \text{ or } (\mathbf{F}, a) \Vdash_V \psi];$$

$$(\mathbf{F}, a) \Vdash_V \varphi \to \psi \iff [not[(\mathbf{F}, a) \Vdash_V \varphi] \text{ or } (\mathbf{F}, a) \Vdash_V \psi];$$

$$(\mathbf{F}, a) \Vdash_V \neg\varphi \iff \text{not } [(\mathbf{F}_C, a) \Vdash_V \varphi];$$

$$(\mathbf{F}, a) \Vdash_V K_i\varphi \iff \forall b \in W[(aR_ib) \implies (\mathbf{F}, b) \Vdash_V \varphi];$$

$$(\mathbf{F}, a) \Vdash_V \Diamond^+\varphi \iff \exists b \in W\,[(aRb) \text{ and } (\mathbf{F}, b) \Vdash_V \varphi];$$

$$(\mathbf{F}, a) \Vdash_V \Diamond^-\varphi \iff \exists b \in W\,[(aR^-b) \text{ and } (\mathbf{F}, b) \Vdash_V \varphi];$$

$$(\mathbf{F}, a) \Vdash_V \mathbf{KnI}\varphi \Leftrightarrow \exists a_{i1}, a_{i2}, \ldots, a_{ik} \in W$$

$$[aR_{i1}a_{i1}R_{i2}a_{i2}\ldots R_{ik}a_{ik}]\&(\mathbf{F}, a_{ik}) \Vdash_V \varphi;$$

$$(\mathbf{F}, a) \Vdash_V \mathrm{U}\varphi \iff [(\mathbf{F}, a) \Vdash_V \mathbf{KnI}\varphi \quad \text{and} \quad (\mathbf{F}, a) \Vdash_V \mathbf{KnI}\neg\varphi];$$

The latter one is an essential step in our approach. So, we assume that a statement φ has uncertain truth value in the current world (state) if agents may, passing to each other information, conclude that φ might be true in some state of the current environment, but that φ can also be false in some state, and this state is also achievable for agents trough a finite transition by agents' accessibility relations.

Now we recall some definitions necessary for description technique applied in the sequel. *Given a model $\mathcal{M} := \langle \mathbf{F}, V\rangle$ based at a frame $\mathbf{F}$ with a base set W and a valuation V, and a formula φ,*

(i) φ is satisfiable *in $\mathcal{M}$ (denotation – $\mathcal{M} \Vdash_{Sat}\varphi$) if there is a state b of $\mathcal{M}$ ($b \in W$) where φ is true:* $(\mathbf{F}, b) \Vdash_V \varphi$.

(ii) φ is valid *in $\mathcal{M}$ (denotation – $\mathcal{M} \Vdash \varphi$) if, for any b of W, the formula φ is true at b ($(\mathbf{F}, b) \Vdash_V \varphi$) w.r.t. V.*

For a frame $\mathbf{F}$ and a formula φ, φ is satisfiable in $\mathbf{F}$ (denotation $\mathbf{F} \Vdash_{Sat}\varphi$) if there is a valuation V in the frame $\mathbf{F}$ such that $\langle \mathbf{F}, V\rangle \Vdash_{Sat}\varphi$. φ is valid in $\mathbf{F}$ (notation $\mathbf{F} \Vdash \varphi$) if *not*$(\mathbf{F} \Vdash_{Sat}\neg\varphi)$.

Definition 1. *The logic* $\mathbf{TS4^U_{K_n}}$ *is the set of all formulas which are valid in all frames* $\mathbf{F}$ *(i.e. valid at all frames w.r.t. all valuations). A formula φ is said to be a theorem of* $\mathbf{TS4^U_{K_n}}$ *if* $\varphi \in \mathbf{TS4_{K_n}}$.

We say a formula φ is *satisfiable* iff there is a valuation V in a Kripke frame $\mathbf{F}$ which makes φ satisfiable: $\langle \mathbf{F}, V\rangle \Vdash_{Sat}\varphi$. Clearly, a formula φ is satisfiable iff

$\neg\varphi$ is not a theorem of $\mathbf{TS4^U_{K_n}}$: $\neg\varphi \notin \mathbf{TS4^U_{K_n}}$, and vice versa, φ is a theorem of $\mathbf{TS4^U_{K_n}}$ ($\varphi \in \mathbf{TS4^U_{K_n}}$) if $\neg\varphi$ is not satisfiable.

3 Decidability of $\mathbf{TS4^U_{K_n}}$

In this section we study computational problems for $\mathbf{TS4^U_{K_n}}$ and describe main technical result of this paper: solution of the decidability and the satisfiability problems for $\mathbf{TS4^U_{K_n}}$. Actually we will use the technique and the scheme of solution used already earlier in our work [29], and the current paper just extend the latter one to handle uncertainty via interaction of agents. Here, as earlier, for technical reason (which makes all constructions much shorter and efficient) we will use transformation or formulas to simple inference rules. Most gain from this transformation is that we will then consider only very simple and uniform formulas - formulas without nested operations (this simplifies the proofs and allows to avoid the necessity to consider nested operations, and hence proofs by induction over formula complexity). First we recall some technical definitions. A (sequential) (inference) rule is an expression (statement)

$$\mathbf{r} := \frac{\varphi_1(x_1,\ldots,x_n),\ldots,\varphi_l(x_1,\ldots,x_n)}{\psi(x_1,\ldots,x_n)},$$

where $\varphi_1(x_1,\ldots,x_n),\ldots,\varphi_l(x_1,\ldots,x_n)$ and $\psi(x_1,\ldots,x_n)$ are formulas constructed out of letters $x_1,\ldots,x_n$. The letters $x_1,\ldots,x_n$ are the variables of $\mathbf{r}$, we use the notation $x_i \in Var(\mathbf{r})$. A meaning of a rule $\mathbf{r}$ is that the statement (formula) $\psi(x_1,\ldots,x_n)$ (which is called conclusion) follows from statements (formulas) $\varphi_1(x_1,\ldots,x_n)$, ..., $\varphi_l(x_1,\ldots,x_n)$ which are called premisses.

Definition 2. *A rule* $\mathbf{r}$ *is said to be* valid *in a Kripke model* $\langle \mathbf{F}, V\rangle$ *(notation* $\mathbf{F} \Vdash_V r$*) if* $[\forall a\ ((\mathbf{F},a) \Vdash_V \bigwedge_{1\le i\le l}\varphi_i)] \Rightarrow \forall a\ ((\mathbf{F},a) \Vdash_V \psi)$. *Otherwise we say* $\mathbf{r}$ *is refuted in* $\mathbf{F}$*, or refuted in* $\mathbf{F}$ *by* V*, and write* $\mathbf{F} \not\Vdash_V \mathbf{r}$. *A rule* $\mathbf{r}$ *is valid in a frame* $\mathbf{F}$ *(notation* $\mathbf{F} \Vdash \mathbf{r}$*) if, for any valuation* V, $\mathbf{F} \Vdash_V \mathbf{r}$

Given a formula φ we can convert it into the rule $x \to x/\varphi$ and employ a technique of reduced normal forms for inference rules as follows. Evidently,

Lemma 1. *A formula* φ *is a theorem of* $\mathbf{TS4^U_{K_n}}$ *iff the rule* $(x \to x/\varphi)$ *is valid in any frame* $\mathbf{F}$.

A rule $\mathbf{r}$ is said to be in *reduced normal form* if $\mathbf{r} = \varepsilon/x_1$ where

$$\varepsilon := \bigvee_{1\le j\le l}\Big(\bigwedge_{1\le i,k\le n, i\ne k}[x_i^{t(j,i,0)} \wedge (\Diamond^+ x_i)^{t(j,i,1)} \wedge (\Diamond^- x_i)^{t(j,i,2)} \wedge \bigwedge_{1\le q\le n}(\neg \mathbf{K}_q \neg x_i)^{t(j,i,q,1)} \wedge \mathbf{KnI} x_i^{t(j,i,3)} \wedge (\mathrm{U} x_i)^{t(j,i,4)}]\Big),$$

all x_s are certain letters (variables), $t(j,i,z), t(j,i,k,z) \in \{0,1\}$ and, for any formula α above, $\alpha^0 := \alpha$, $\alpha^1 := \neg\alpha$.

Definition 3. *Given a rule* $\mathbf{r_{nf}}$ *in reduced normal form,* $\mathbf{r_{nf}}$ *is said to be a normal reduced form for a rule* $\mathbf{r}$ *iff, for any frame* $\mathbf{F}$, $\mathbf{F} \Vdash \mathbf{r} \Leftrightarrow \mathbf{F} \Vdash \mathbf{r_{nf}}$.

Reasoning by the same scheme of proof as in Lemma 3.1.3 and Theorem 3.1.11 from [22] we obtain

Theorem 1. *There exists an algorithm running in (single) exponential time, which, for any given rule* $\mathbf{r}$*, constructs its normal reduced form* $\mathbf{r_{nf}}$.

For readers interested in details of this technique we put below a draft of proof for Theorem 1. Actually we shall specify the general algorithm described in Lemma 3.1.3 and Theorem 3.1.11 [22] to the language of our logic.

Assume we are given with a rule

$$\mathbf{r} = \frac{\varphi_1(x_1, ..., x_n), ..., \varphi_m(x_1, ..., x_n)}{\psi(x_1, ..., x_n)}$$

It is evident that $\mathbf{r}$ is equivalent to the rule

$$\mathbf{r_0} = \frac{\varphi_1(x_1, ..., x_n) \wedge ... \wedge \varphi_m(x_1, ..., x_n) \wedge x_c \equiv \psi(x_1, ..., x_n)}{x_c}$$

where x_c is a new variable. Therefore we can restrict the case to the rules in the form $\mathbf{c} = \varphi(x_1, ..., x_n)/x_c$.

If $\varphi = \alpha \circ \beta$, where $\circ$ is a binary logical operation and both formulas α and β are not simply variables or unary logical operations applied to variables (which both we call final formulas), take two new variables x_α and x_β and the rule

$$\mathbf{r_1} := (x_\alpha \circ x_\beta) \wedge (x_\alpha \equiv \alpha) \wedge (x_\beta \equiv \beta)/x_c.$$

If one from formulas α or β is final and another one not, we apply this transformation to the non-final formula. It is clear that $\mathbf{r}$ and $\mathbf{r_1}$ are equivalent w.r.t. validity in frames.

If $\varphi = *\alpha$, where $*$ is a unary logical operation and α is not a variable, take a new variable x_α and the rule

$$\mathbf{r_1} := *x_\alpha \wedge (x_\alpha \equiv \alpha)/x_c.$$

Again $\mathbf{r}$ and $\mathbf{r_1}$ are equivalent. We continue this transformation over the resulting rules

$$\frac{\bigwedge_{j \in J_1} \gamma_j \wedge \bigwedge_{i \in I_1} x_{\alpha_i} \equiv \alpha_i}{x_c}$$

until all formulas α_i and γ_j in the premise of the resulting rules will be either atomic formulas, i.e. logical operations applied to variables, or variables. Evidently this transformation is *polynomial*. Further, we transform the premise of the resulting rule in the disjunctive normal form and make disjunctive normal form to be perfect (having the disjunctive members to be uniform length and containing all the components required in the definition of reduced normal forms)

and obtain as the result an equivalent rule $\mathbf{r_2}$. This transformation, as well as all known ones for reduction of Boolean formulas to disjunctive normal forms, is *exponential* in time. As the result the final rule $\mathbf{r_f}$ has the required form. This concludes the proof. □

Recall now that the decidability of our logic (in particular decidability of the satisfiability problem) will follow (by this theorem) if we find an algorithm recognizing rules in reduced normal form which are valid in all frames $\mathbf{F}$. As in earlier works, most important starting point is to develop a technique to handle interactions of agents. This technical step is carried out in the following lemma (which, as earlier, might be proved using a tick similar to the one used in proof of Lemma 8 in Rybakov [26]).

Lemma 2. *A rule* $\mathbf{r_{nf}}$ *in reduced normal form is refuted in a frame* $\mathbf{F}$ *if and only if* $\mathbf{r_{nf}}$ *can be refuted in a frame with time clusters of size square exponential from* $\mathbf{r_{nf}}$.

Based at this lemma, and applying then a technique developed from standard filtration technique, we may prove

Lemma 3. *A rule* $\mathbf{r_{nf}}$ *in reduced normal form is refuted in a frame* $\mathbf{F}$ *iff* $\mathbf{r_{nf}}$ *can be refuted in a finite frame* $\mathbf{F}_1$ *by a special valuation* V*, where the size of the frame* $\mathbf{F}_1$ *has effective upper bound computable from the size of* $\mathbf{r_{nf}}$.

Using Theorem 1, Lemma 1 and Lemma 3 we derive our main result:

Theorem 2. *The logic* $\mathbf{TS4^U_{K_n}}$ *is decidable; the satisfiability problem for* $\mathbf{TS4^U_{K_n}}$ *is decidable.*

Possible Applications: The technique of this paper and obtained algorithm may be applied in research analyzing statements about multi-agent reasoning. E.g. verification '*if a statement (specification) is consistent (i.e. satisfiable)*' may be performed via suggested algorithm. Though computational complexity of this algorithm is high, the verification yet might be (in many cases) done by direct inspection via construction of a frame disproving a rule (satisfying the formula), where the frame could have permissible size. Besides, suggested framework may clarify essence of reasoning via analyzing interdependence of statements with logically oriented substance.

4 Conclusion, Future Work

This paper constructs a framework to study logical properties of reasoning in multi-agents' environment and to construct tools for computation satisfiable and valid statements. Main aim here is to model logical uncertainty via interaction of agents. We suggest the logic $\mathbf{TS4^U_{K_n}}$ based at non-linear frames describing this approach and show that $\mathbf{TS4^U_{K_n}}$ is decidable (and hence the satisfiability problem for this logic is also decidable). Suggested algorithms are based at computation

refutability of rules in reduced from at special finite frames of effectively bounded size. Developed approach is rather flexible and allows to work with a variety of logics from AI and CS.

Future research in suggested line may include investigation of AI-logics describing reasoning modeled by frames based at non-transitive time, which may reflect computations (transitions) with bounded introspection (to future and past). Aiomatizability for $\mathbf{TS4^{U}_{K_n}}$ is an open problem, besides complexity issues and possible ways of refining the complexity bounds in the suggested algorithm are also interesting. Problems of decidability w.r.t. admissible inference rules for $\mathbf{TS4^{U}_{K_n}}$ and similar logics are not investigated yet. Interesting direction is to model logical uncertainty via different truth values of the statements in 'neighborhoods' of a current state (but not merely in current time cluster, as in this paper). For example, this might be actual in the case of intransitive logics, where only 'tomorrow' and 'yesterday' clusters are taken to consideration.

References

1. Arisha, K., Ozcan, F., Ross, R., Subrahmanian, V.S., Eiter, T., Kraus, S.I.: A platform for collaborating agents. IEEE Intelligent Systems 14(2), 64–72 (1999)
2. Avouris, N.M.: Co-operation knowledge-based systems for environmental decision-support. Knowledge-Based Systems 8(1), 39–53 (1995)
3. Babenyshev, S., Rybakov, V.: Logic of Plausibility for Discovery in Multi-agent Environment Deciding Algorithms. In: Falsafi, B., VijayKumar, T.N. (eds.) PACS 2000. LNCS, vol. 2008, pp. 210–217. Springer, Heidelberg (2001)
4. Babenyshev, S., Rybakov, V.: Decidability of Hybrid Logic with Local Common Knowledge Based on Linear Temporal Logic LTL. In: Beckmann, A., Dimitracopoulos, C., Löwe, B. (eds.) CiE 2008. LNCS, vol. 5028, pp. 32–41. Springer, Heidelberg (2008)
5. Babenyshev, S., Rybakov, V.: Logic of Discovery and Knowledge: Decision Algorithm. In: Falsafi, B., VijayKumar, T.N. (eds.) PACS 2000. LNCS, vol. 2008, pp. 711–718. Springer, Heidelberg (2001)
6. Babenyshev, S., Rybakov, V.: Describing Evolutions of Multi-Agent Systems. In: Velásquez, J.D., Ríos, S.A., Howlett, R.J., Jain, L.C. (eds.) KES 2009, Part I. LNCS, vol. 5711, pp. 38–45. Springer, Heidelberg (2009)
7. Brachman, R.J., Schmolze, J.G.: An overview on the KL-ONE knowledge representation system. Cognitive Science 9(2), 179–226 (1985)
8. Barwise, J.: Three Views of Common Knowledge. In: Vardi (ed.) Proc. Second Confeuence on Theoretical Aspects of Reasoning about Knowledge, pp. 365–379. Morgan Kaufmann, San Francisco (1988)
9. Dwork, C., Moses, Y.: Knowledge and Common Knowledge in a Byzantine Environment: Crash Failures. Information and Computation 68(2), 156–183 (1990)
10. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning About Knowledge, p. 410. The MNT Press, Cambridge (1995)
11. Hendler, J.: Agents and the semantic web. IEEE Intelligent Systems 16(2), 30–37 (2001)
12. Kifer, M., Lozinski, L.: A Logic for Reasoning with Inconsistency. J. Automated Deduction 9, 171–215 (1992)

13. Kraus, S., Lehmann, D.L.: Knowledge, Belief, and Time. Theoretical Computer Science 98, 143–174 (1988)
14. Moses, Y., Shoham, Y.: Belief and Defeasible Knowledge. Artificial Intelligence 64(2), 609 - 322 (1993)
15. Nebel, B.: Reasoning and Revision in Hybrid Representation Systems. LNCS, vol. 322 (1940)
16. Neiger, G., Tuttle, M.R.: Common knowledge and consistent simultaneous coordination. Distributed Computing 5(3), 334–352 (1993)
17. Nguyen, N.T., Jo, G.-S., Howlett, R.J., Jain, L.C. (eds.): KES-AMSTA 2008. LNCS (LNAI), vol. 4953. Springer, Heidelberg (2008)
18. Nguyen, N.T., Huang, D.S.: Knowledge Management for Autonomous Systems and Computational Intelligence. Journal of Universal Computer Science 15(4) (2009)
19. Nguyen, N.T., Katarzyniak, R.: Actions and Social Interactions in Multi-agent Systems. Special issue for International Journal of Knowledge and Information Systems 18(2) (2009)
20. Sakama, C., Son, T.C.: Interactind Answer Set. In: Dix, J., Fisher, M., Novák, P. (eds.) CLIMA X. LNCS, vol. 6214, pp. 122–140. Springer, Heidelberg (2010)
21. Rybakov, V.V.: A Criterion for Admissibility of Rules in the Modal System S4 and the Intuitionistic Logic. Algebra and Logic 23(5), 369–384 (1984) (Engl. Translation)
22. Rybakov, V.V.: Admissible Logical Inference Rules. Studies in Logic and the Foundations of Mathematics, vol. 136. Elsevier Sci. Publ., North-Holland (1997)
23. Rybakov, V.V.: Logical Consecutions in Discrete Linear Temporal Logic. Journal of Symbolic Logic (ASL, USA) 70(4), 1137–1149 (2005)
24. Rybakov, V.: Until-Since Temporal Logic Based on Parallel Time with Common Past. Deciding Algorithms. In: Artemov, S., Nerode, A. (eds.) LFCS 2007. LNCS, vol. 4514, pp. 486–497. Springer, Heidelberg (2007)
25. Rybakov, V.: Logic of knowledge and discovery via interacting agents - Decision algorithm for true and satisfiable statements. Inf. Sci. 179(11), 1608–1614 (2009)
26. Rybakov, V.: Linear Temporal Logic LTK_K extended by Multi-Agent Logic K_n with Interacting Agents. J. Log. Comput. 19(6), 989–1017 (2009)
27. Rybakov, V.V.: Representation of Knowledge and Uncertainty in Temporal Logic LTL with Since on Frames Z of Integer Numbers. In: König, A., Dengel, A., Hinkelmann, K., Kise, K., Howlett, R.J., Jain, L.C. (eds.) KES 2011, Part I. LNCS, vol. 6881, pp. 306–315. Springer, Heidelberg (2011)
28. Rybakov, V.V.: Agents' Logics with Common Knowledge and Uncertainty: Unification Problem, Algorithm for Construction Solutions. In: König, A., Dengel, A., Hinkelmann, K., Kise, K., Howlett, R.J., Jain, L.C. (eds.) KES 2011, Part I. LNCS, vol. 6881, pp. 171–179. Springer, Heidelberg (2011)
29. Rybakov, V.: Multi-Agent Logic based on Temporary Logic $TS4_{K_n}$ serving Web Search. In: Grana, M., et al. (eds.) KES 2012, Advances in Knowledge-Based and Intelligent Information and Engineering Systems. Frontiers in Artificial Intelligence and Applications, vol. 243, pp. 108–117 (2012)
30. Rychtyckyi, N.: DLMS: An evaluation of KL-ONE in the automobile industry. In: Aiello, L.C., Doyle, J., Shapiro, S. (eds.) Proc. of he 5-th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 1996), pp. 588–596. Morgan Kaufmann, San Francisco (1996)

Opponent Modelling by Sequence Prediction and Lookahead in Two-Player Games

Richard Mealing and Jonathan L. Shapiro

University of Manchester, Manchester M13 9PL, UK
{mealingr,jls}@cs.man.ac.uk

Abstract. Learning a strategy that maximises total reward in a multi-agent system is a hard problem when it depends on other agents' strategies. Many previous approaches consider opponents which are reactive and memoryless. In this paper, we use sequence prediction algorithms to perform opponent modelling in two-player games, to model opponents with memory. We argue that to compete with opponents with memory, lookahead is required. We combine these algorithms with reinforcement learning and lookahead action selection, allowing them to find strategies that maximise total reward up to a limited depth. Experiments confirm lookahead is required, and show these algorithms successfully model and exploit opponent strategies with different memory lengths. The proposed approach outperforms popular and state-of-the-art reinforcement learning algorithms in terms of learning speed and final performance.

Keywords: Opponent Modelling, Sequence Prediction, Lookahead, Reinforcement Learning, Multi-Agent Learning, Game Theory.

1 Introduction

A problem an agent in a multi-agent system often faces is how to maximise the total of their rewards given these depend on other agents' strategies. This is challenging because the reward maximising strategy will typically change as other agents change their strategies. Additionally other agents' strategies are usually private. Many algorithms assume other agents' strategies are memoryless and only depend on the environment's state. However, intelligent agents will generally change their strategies depending on some memory of previous actions. We focus on maximising total reward in competitive two-player games and assume the opponent selects actions based on a sequential memory of previous actions. This is true for two opponent types we consider: those which have a memory of their own actions, and those which have a memory of both their own and the agent's actions. These opponents arguably better resemble human-like strategies that react based on a short-term memory as opposed to sampling from a probability distribution.

Reinforcement learning algorithms, such as Q-learning [1], do not explicitly model the opponent's strategy separately from the environment. Traditional opponent modelling algorithms, such as fictitious play [2], ignore the possibility

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 385–396, 2013.

that the opponent's strategy is based on a sequential memory. However, playing optimally against such an opponent requires the agent to have their own memory to learn the opponent's strategy, but also to look ahead to find actions that will maximise their total reward. Our proposed solution has three parts:

1. We use sequence prediction algorithms to model opponents with a sequential memory. We modify them to make opponent action predictions given a (possibly empty) sequence of hypothesised actions.
2. We combine our sequence prediction opponent modelling (SP-OM) algorithms with a reinforcement learning algorithm (Q-learning [1]) to learn the (possibly delayed) values of the agent's own actions for each game state.
3. We use exhaustive explicit lookahead with the learnt opponent model and action values to greedily select future action sequences from all possible paths (up to a limited depth) that maximise total reward.

We compare our proposed approach to alternative approaches by using a variety of reinforcement learning algorithms that observe game states augmented with a memory of previous opponent and/or player actions. Instead of explicit lookahead these algorithms use discounting as an implicit form of lookahead. Experiments show that our proposed approach generally gains higher average rewards at a faster rate compared to these alternatives.

2 Related Work

Opponent modelling research tends to focus on specific games (e.g. chess, poker, go, etc) limiting the use of resulting algorithms. However, there have been some general approaches. Fictitious play is one of the earliest and most popular opponent modelling algorithms. It estimates an opponent's strategy by normalising counts of its actions in each game state. Originally proposed by Brown in 1951 to explain Nash equilibrium play [2] it is still used and has many extensions. Its main problem is that it assumes a fixed opponent strategy that has one state per game state, which is not always the case. For example, tit-for-tat is a two-state strategy for playing the single-state prisoner's dilemma game.

Carmel and Markovitch [3] assumed agents' strategies could be modelled as deterministic finite automata and presented an online heuristic algorithm that can infer these models given input/output behaviour. Their Unsupervised L* (US-L*) algorithm learns very compact models with high accuracy. Its main problem is its high computational complexity.

Jensen et al. [4] proposed a sequence prediction method called ELPH (Entropy Learned Pruned Hypothesis space). ELPH forms hypotheses over possible conditional probability distributions based on interaction histories of different lengths and predicts using those with the lowest entropy. It can adapt rapidly against non-stationary strategies and can defeat human and agent players in rock-paper-scissors. ELPH is one of the methods we use in this paper.

Knoll and de Freitas [5] examined a state-of-the-art compression method called PAQ version 8L (PAQ8L). They used its sequence prediction capabilities to predict opponent actions in rock-paper-scissors and reported that it usually won

human players. They explained that this is because humans typically used predictable patterns after a large number of rounds. Like US-L*, one of its main problems is a high computational complexity.

3 Background

3.1 Games in Our Experiments

The first two experiments (sections 6.1 and 6.2) use two-player iterated matrix games with the payoff matrices shown in Tables 1a and 1b respectively. In a two-player matrix game the row and column players select actions i.e. a row i and a column j simultaneously. Each player receives the reward in their payoff matrix at (i, j). In an iterated version the game is played repeatedly; the goal is to accumulate the highest possible reward.

Table 1. Matrix games experiments 6.1 and 6.2 use with each matrix entry (p_1, p_2) showing payoffs for the row player and the column player respectively.

(a) Rock-Paper-Scissors

	R	P	S
R	0,0	-1,1	1,-1
P	1,-1	0,0	-1,1
S	-1,1	1,-1	0,0

(b) Prisoner's Dilemma

	D	C
D	1,1	4,0
C	0,4	3,3

In iterated rock-paper-scissors, it is possible to win more games in expectation unless the opponent is choosing actions uniformly at random. Humans are known to be poor random number generators [6] and thus exploitable in this game. This has seeded interest in many human and algorithmic rock-paper-scissors tournaments. We use iterated rock-paper-scissors in experiment 6.1 to show a memory is needed to respond optimally to an opponent whose strategy depends on a memory of their previous actions. The iterated prisoner's dilemma is named so because defection gives the highest reward independent of the opponent's action but if both defect they get less than if both cooperate. Interest in this game was promoted by Axelrod's tournament [7] where he found altruistic cooperative strategies outperformed greedy defecting strategies. We use the iterated prisoner's dilemma in experiment 6.2 to show lookahead is necessary to maximise total reward against an opponent whose strategy depends on a memory of their own and the agent's previous actions.

The third experiment (section 6.3) uses Littman's soccer game [8]. It is played on a 4×5 grid with goals either side. Each player can move north, south, east, west or they can stand still and must dribble the ball into their opponent's goal. Each game starts with the ball being given to a random player. At each step, players select actions, which are executed randomly adding non-determinism. Scoring (being scored against) gets a reward of 1 (-1) and resets players to their

initial positions. Moving to an occupied spot fails and gives the ball (if possible) to the opponent. We use Littman's soccer game in experiment 6.3 to compare our approach to popular and state-of-the-art reinforcement learning algorithms on a larger and extensive-form (non-matrix) game.

3.2 Sequence Prediction Opponent Modelling

Sequence prediction is one approach in the broader field of context prediction. For an overview of context prediction methods see [9]. Our selected sequence prediction methods can observe a sequence of symbols, also called a context, $x^{t-n+1}, x^{t-n+2}, ..., x^t$ from some alphabet $x^i \in \Sigma$ where $(t-n+1) \leq i \leq t$, t is time and n is the (possibly unbounded) short-term memory size, and predict the next observation x^{t+1}. Probabilistic predictions can also be returned based on previous observations i.e. $\Pr\left(x^{t+1}|x^t, x^{t-1}, ..., x^{t-n}\right)$. Our work is an example of Markov process analysis where probabilistic prediction algorithms are used to model Markov processes. Lempel-Ziv-1978 (LZ78) and Knuth-Morris-Pratt require some modification to achieve probabilistic predictions. LZ78 is modified as follows: it finds all matching contexts to the current context with one extra symbol. The returned probability distribution is then each longer context's extra symbol with a probability equal its context's count divided by the current context's count. Knuth-Morris-Pratt returns probabilistic predictions by finding the longest matches to the end of its string of observations. The returned probability distribution is the normalised counts for each symbol that occurred after the string's last symbol in each of the longest matches. Furthermore, we modified all methods to be able to return probabilistic predictions given a hypothesised context i.e. $\Pr\left(x^{t+k}|x^{t+k-1}, x^{t+k-2}, ..., x^{t+k-n}\right)$ where $k \geq 1$ is the amount of lookahead and x^i is a hypothesised symbol if $i > t$. For each method this required saving its current parameters, observing the hypothesised sequence, returning probabilistic predictions as usual and finally restoring its saved parameters. This ensured unobserved sequences did not update the opponent model.

3.3 Q-Learning

We combine Watkins Q(uality)-learning algorithm [1] with our SP-OM algorithms to learn the (possibly delayed) values of the agent's own actions in each game state. Watkins showed that Q-learning attempts to maximise the sum of expected future discounted rewards. Formally it learns an action-value function that outputs the expected value of taking action a in state s and following a fixed strategy thereafter. It can be expressed by equation 1.

$$Q\left(s^t, a^t\right) \leftarrow (1-\alpha)\, Q\left(s^t, a^t\right) + \alpha \left[r + \gamma \max_{a^{t+1}} Q\left(s^{t+1}, a^{t+1}\right)\right] \tag{1}$$

where s^t is the state at time t, a^t is the action at time t, $0 \leq \alpha \leq 1$ is the learning rate, r is the reward for taking action a^t in state s^t and $0 \leq \gamma \leq 1$ is the discount factor, which controls the amount of implicit lookahead (see section 3.4).

3.4 The Need for Lookahead

We use explicit lookahead with our SP-OM algorithm's opponent action predictions and Q-learning's estimates of the agent's own action values to directly find the future path(s) with the highest total reward. The amount of lookahead is limited as search time grows exponentially with it. To show the necessity of lookahead consider the iterated prisoner's dilemma. With lookahead of 1 the agent would always defect since it has the highest reward for the next time step irrespective of the opponent's strategy. With lookahead of 2 defecting twice only gives the highest total reward if the opponent cooperates twice despite the initial defection, which is unlikely against an intelligent opponent. If instead the opponent reacts by copying the agent's previous move (i.e. the tit-for-tat strategy) then the agent would increase their total reward by cooperating on the first move and defecting on the second. Thus, using the payoffs in Table 1b, lookahead of 1 would defect at each time-step giving t total reward whereas lookahead of 2 would cooperate giving $3t$.

Q-learning's discounting in equation 1 can be used as implicit lookahead. If the states are defined as opponent actions then a discount factor of $\gamma = 0$ would learn the payoff matrix and selecting an action via $\arg\max_{a_i} Q\left(a_{\text{opp}}^{t+1}, a_i\right)$ would be the same as selecting an action with an explicit lookahead of 1. For $\gamma > 0$ there is more lookahead because, it estimates the value an opponent action player action pair as the payoff matrix value added to a proportion of the maximum value for the next opponent action. This proportion increases as γ increases. Only with $\gamma = 1$ is sufficient lookahead guaranteed.

3.5 Fictitious Play

We use fictitious play (FP) as a comparison opponent modelling algorithm to our SP-OM algorithms. It assumes that for each game state the opponent randomly samples from a fixed probability distribution over their actions [2]. It estimates this distribution by normalising counts of the opponent's actions via equation 2.

$$\pi_{\text{opp}}^{t}\left(s^t, a_{\text{opp}}\right) \leftarrow \pi_{\text{opp}}^{t-1}\left(s^t, a_{\text{opp}}\right)\left(1 - \frac{1}{t}\right) + \frac{1}{t}\mathbb{I}\left(a_{\text{opp}} = a_{\text{opp}}^{t}\right) \tag{2}$$

where π_{opp}^{t} is the estimated opponent strategy at time t, s^t is the state at time t and $\pi_{\text{opp}}^{t}\left(s^t, a_{\text{opp}}\right)$ returns the probability of the opponent's action a_{opp}.

4 Sequence Prediction

Generally each method updates its long-term memory by getting distributions associated with its short-term memory and updating them with an observation. Its short-term memory is then updated to include the observation and distributions associated with the new short-term memory are used to make predictions. Our selected sequence prediction methods include:

- **Lempel-Ziv-1978 (LZ78)** [**10**] and **Knuth-Morris-Pratt (KMP)** [**11**], which have unbounded context lengths.
- **Prediction by Partial Matching version C (PPMC)** [**12**] and **ActiveLeZi** [**13**], which blend predictions from different context lengths.
- **Transition Directed Acyclic Graph (TDAG)** [**14**] and **Entropy Learned Pruned Hypothesis Space (ELPH)** [**4**], which remove unlikely contexts.
- **N-Gram** [**15**] and **Hierarchical N-Gram (H. N-Gram)** [**15**], where the latter is a group of 1 to N-Grams and predicts using the longest context.
- **Long Short Term Memory (LSTM)** [**16**], which is a recurrent neural network and implicitly incorporates prediction blending and context pruning.

5 Effective Responses Using Lookahead

We use lookahead with learnt action values and opponent action predictions to maximise total reward via algorithm 1. The scalability of our approach depends on the complexities of Q-Learning, the prediction algorithm and the lookahead technique. In large domains the former two can be mitigated by abstraction. In our experiments exact exhaustive lookahead dominates and is bounded by $\sum_{t=0}^{T} |A|^t$ where $|A|$ is the maximum number of actions in a state and T is the lookahead depth. For deep lookahead other approaches are needed (future work). In experiments 6.1 and 6.2 observations are joint actions $x^t = (a^t_{\text{opp}}, a^t_i)$ and state transitions are deterministic $\Pr(s^t) = 1$. In experiment 6.3 observations are opponent actions $x^t = a^t_{\text{opp}}$ and state transitions are probabilistic $\Pr(s^t) = 0.5$ if players try to move to the same place otherwise they are deterministic $\Pr(s^t) = 1$.

Algorithm 1. Player i using prediction and Q-learning with T step lookahead

Require: Q-function $Q : S \times A_i \to \mathbb{R}$, Previous state $s^{t-1} \in S$, Previous player action $a_i^{t-1} \in A_i$, Learning rate α, Reward r, Discount factor γ, Current state $s^t \in S$, Prediction algorithm P_{alg} (short-term memory size n), Current observation x^t, Lookahead T

1: Update the q-value

$$Q\left(s^{t-1}, a_i^{t-1}\right) \leftarrow (1-\alpha)\, Q\left(s^{t-1}, a_i^{t-1}\right) + \alpha \left[r + \gamma \max_{a_i^t} Q\left(s^t, a_i^t\right) \right]$$

2: Observe x^t with the prediction algorithm P_{alg}

3: Find the action sequence with the maximum total reward z^*

$$z^* \leftarrow \arg \max_{a_i^{t+1:t+T}} \left[\sum_{j \leftarrow 1}^{T} \left(\prod_{j' \leftarrow 1}^{j} \Pr\left(s^{t+j'}\right) \underbrace{\Pr\left(a^{t+j'}_{\text{opp}} | x^{(t+j'-1):(t+j'-n)}\right)}_{\text{predicted by } P_{\text{alg}}} \right) Q\left(s^{t+j}, a_i^{t+j}\right) \right]$$

where $y^{i:j} = \left(y^i, y^{i+1}, ..., y^{i+j}\right)$ assuming $i < j$

4: **return** a_i^{t+1} from z^*

6 Results

All experiments compare our approach using algorithm 1 and different opponent modelling algorithms (i.e. either fictitious play, see section 3.5, or our SP-OM versions of the algorithms listed in section 4) to the following reinforcement learning algorithms: an ϵ-greedy Q-Learner, Win or Learn Fast Policy Hill Climbing (WoLF-PHC) [17], Weighted Policy Learner (WPL) [18] and Policy Gradient Ascent with Approximate Policy Prediction (PGA-APP) [19]. Table 2 shows all algorithm parameters for all games. In experiments 6.1 and 6.2 memory size n means remembering the previous n opponent and n player actions. In experiment 6.3 it means remembering the previous n opponent actions. Note in all tables lighter gradients indicate better values, N/A refers to an unbounded memory length and SEM means standard error of the mean.

Table 2. Parameters for all algorithms in all games. For Littman's soccer game our parameters are based on his [8]. Lookahead and short-term memory size are reduced in the soccer game to make it more tractable.

	Iter. rock-paper-scissors	Iter. prisoner's dilemma	Littman's soccer
Training exploration rate	$\min\left(1, \frac{1}{3\times 10^{-2}t}\right)$	$\min\left(1, \frac{1}{3\times 10^{-2}t}\right)$	0.20
Testing exploration rate	0	0	0
Training steps	1×10^4	1×10^4	1×10^5
Testing steps	1×10^3	1×10^3	1×10^5
Runs	100	100	500
Prediction and reinforcement learning algorithms			
Learning rate	0.99	0.99	$\left(10^{\frac{\log 0.01}{T}}\right)^t$
Prediction algorithms			
Memory size	1-3	1-4	1
Lookahead	2	1, 2, 2	1
Discount factor	0	0, 0, 0.99	0.9
Reinforcement learning algorithms			
Discount factor	0.99	0, 0.99, 0.99	0.9
WoLF-PHC, WPL and PGA-APP			
Step-size	0.05	0.05	$\left(10^{\frac{\log 0.01}{T}}\right)^t$
WoLF-PHC			
Winning step-size	0.05	0.05	$\left(10^{\frac{\log 0.01}{T}}\right)^t$
Losing step-size	0.10	0.10	$2\left(10^{\frac{\log 0.01}{T}}\right)^t$
PGA-APP			
Prediction length	1	1	1

6.1 Iterated Rock-Paper-Scissors

This experiment aims to show memory is needed to best respond to an opponent whose strategy depends on their previous actions and to compare the performances of algorithm 1 and the reinforcement learning algorithms against these opponents. We played algorithm 1, using each SP-OM algorithm and fictitious play in turn as its prediction algorithm, as well as the reinforcement learning algorithms against opponents playing deterministic action sequences repeatedly (e.g. R,P,S,R,P,S,...). Table 3 shows agents cannot learn to play best response

(reach an average payoff per step of 1, win every game) if their memory is less than the opponent's model order (right of the dividing line) as they cannot know the opponent's next action. If their memory is at least as great as the opponent's model order (left of the dividing line), they can learn to play best response. Table 3 also shows for all memory sizes and model orders (except memory 1 order 2 and memory 2 order 3) algorithm 1 with SP-OM algorithms, gains the highest payoffs at generally the fastest rates (i.e. has the lowest average times to reach payoffs). LSTM performs poorly but, has many parameters (e.g. initial weights, activation functions, etc), which if further tuned might improve its performance.

6.2 Iterated Prisoner's Dilemma

This experiment aims to show lookahead is necessary to obtain high total payoffs against an opponent whose strategy depends on their own and the agent's previous actions and to compare the performances of algorithm 1 and the reinforcement learning algorithms against these opponents. We played algorithm 1, using each SP-OM algorithm and fictitious play in turn as its prediction algorithm, and the reinforcement learning algorithms against 20 finite automata (described in [20]) in separate iterated prisoner's dilemma tournaments. Each tournament's players all faced each other without self-play. Table 4a shows reinforcement learning algorithms with a discount factor (implicit lookahead) of 0 and algorithm 1 with (explicit) lookahead of 1 have average payoffs per step over all memories of 1.93 and 1.93. Table 4b shows increasing the discount factor to 0.99 and the lookahead to 2 increases these values to 2.32 and 2.62 respectively. This is because the increased lookahead allows agents to consider opponent reactions to their actions. Table 4c shows giving algorithm 1 both a lookahead of 2 and a discount factor of 0.99 increases its value further to 2.67 but with an increased average time over all memories of 96 steps compared to 31 steps with just a lookahead of 2. The tables show using algorithm 1 with lookahead 2 and SP-OM algorithms that have sufficient memory gains the highest payoffs at generally the fastest rates. For example in table 4b the highest payoff of 2.873 (1st place) after 60 steps is achieved by LZ78 whereas for a reinforcement learning algorithm (ϵ-greedy Q-Learning) the highest is 2.74 (1st place) after 180 steps.

6.3 Soccer

This experiment aims to directly compare algorithm 1 to the reinforcement learning algorithms on a larger, extensive-form game. We played algorithm 1, using each SP-OM algorithm and fictitious play in turn as its prediction algorithm, against the reinforcement learning algorithms in Littman's soccer game [8]. For algorithm 1, a copy was created in each game state (defined by player positions and ball possession). Table 5 shows algorithm 1 wins above 50% of the games on average using any prediction algorithm. Despite its simplicity fictitious play does surprisingly well but PPMC gives the highest performances in all cases.

Table 3. Shows the performance of algorithm 1, using each SP-OM algorithm and fictitious play in turn as its prediction algorithm, and the reinforcement learning algorithms against opponents playing deterministic action sequences repeatedly. Agent memory size is shown vertically increasing downward. The opponent's action sequence is shown horizontally with model order increasing rightward. Avg Payoff is the average payoff (equal to games won in this case) per step over 1×10^3 testing steps (± SEM). Avg Time is the average number of training steps required to reach the Avg Payoff (± SEM). The dark line divides players with (in)sufficient memory to model the opponent to the (right) left. Maximum allowed time to reach an Avg Payoff was 210 training steps so times around this value may be larger.

	{R,P,S} Order 1			{R,R,P,P,S,S} Order 2			{R,R,R,P,P,P,S,S,S} Order 3		
	Name	Avg Payoff	Avg Time	Name	Avg Payoff	Avg Time	Name	Avg Payoff	Avg Time
Memory Size 1	ELPH	1 ± 0	14.7 ± 0.6	WoLF-PHC	0.645 ± 0.006	89 ± 5	N-Gram	0.667 ± 0	10 ± 0
	H. N-Gram	1 ± 0	15.3 ± 0.7	PGA-APP	0.644 ± 0.008	59 ± 5	TDAG	0.667 ± 0	10 ± 0
	PPMC	1 ± 0	15.6 ± 0.6	ε Q-Learner	0.635 ± 0.008	22 ± 3	H. N-Gram	0.667 ± 0	11 ± 0.3
	TDAG	1 ± 0	15.9 ± 0.6	PPMC	0.627 ± 0.004	89 ± 8	PPMC	0.667 ± 0	17.7 ± 0.6
	N-Gram	1 ± 0	16.8 ± 0.7	N-Gram	0.622 ± 0.003	46 ± 7	ActiveLeZi	0.667 ± 0	19.2 ± 0.6
	ActiveLeZi	1 ± 0	18.9 ± 0.8	TDAG	0.621 ± 0.003	33 ± 5	ELPH	0.666 ± 0.0003	56 ± 4
	WoLF-PHC	1 ± 0	27 ± 2	H. N-Gram	0.618 ± 0.003	46 ± 7	PGA-APP	0.652 ± 0.005	62 ± 4
	PGA-APP	0.973 ± 0.009	24 ± 2	ELPH	0.617 ± 0.002	210 ± 0	WoLF-PHC	0.646 ± 0.004	71 ± 4
	ε Q-Learner	0.97 ± 0.01	29 ± 2	ActiveLeZi	0.613 ± 0.003	53 ± 7	ε Q-Learner	0.582 ± 0.008	48 ± 6
	WPL	0.87 ± 0.01	74 ± 6	WPL	0.374 ± 0.007	143 ± 7	WPL	0.393 ± 0.008	139 ± 7
	LSTM	0.05 ± 0.04	83 ± 7	LSTM	0.001 ± 0.0004	202 ± 4	LSTM	0 ± 0.0001	194 ± 5
Memory Size 2	ELPH	1 ± 0	10 ± 0	N-Gram	1 ± 0	10 ± 0	WoLF-PHC	0.68 ± 0.01	173 ± 6
	PPMC	1 ± 0	15.3 ± 0.6	TDAG	1 ± 0	10 ± 0	TDAG	0.675 ± 0.0009	10.5 ± 0.2
	TDAG	1 ± 0	15.4 ± 0.6	ELPH	1 ± 0	10 ± 0	ActiveLeZi	0.674 ± 0.0009	22 ± 1
	H. N-Gram	1 ± 0	16 ± 0.6	H. N-Gram	1 ± 0	41 ± 1	PPMC	0.673 ± 0.0009	15.2 ± 0.9
	ActiveLeZi	1 ± 0	17.6 ± 0.7	PPMC	1 ± 0	61 ± 1	H. N-Gram	0.673 ± 0.0009	15.8 ± 0.9
	N-Gram	1 ± 0	19 ± 1	ActiveLeZi	1 ± 0	64 ± 1	N-Gram	0.668 ± 0.002	72 ± 4
	WoLF-PHC	0.98 ± 0.008	91 ± 3	ε Q-Learner	0.92 ± 0.01	45 ± 4	ε Q-Learner	0.64 ± 0.01	56 ± 5
	ε Q-Learner	0.97 ± 0.01	28 ± 2	WoLF-PHC	0.91 ± 0.01	147 ± 8	PGA-APP	0.61 ± 0.01	120 ± 7
	PGA-APP	0.92 ± 0.01	52 ± 3	PGA-APP	0.86 ± 0.01	109 ± 6	ELPH	0.6 ± 0.002	58 ± 4
	WPL	0.65 ± 0.02	105 ± 7	WPL	0.54 ± 0.01	71 ± 6	WPL	0.375 ± 0.009	139 ± 7
	LSTM	-0.04 ± 0.03	115 ± 8	LSTM	0.016 ± 0.0003	210 ± 0	LSTM	-0.003 ± 0.003	118 ± 7
Memory Size 3	ELPH	1 ± 0	10 ± 0	TDAG	1 ± 0	10.1 ± 0.1	TDAG	1 ± 0	10.1 ± 0.1
	PPMC	1 ± 0	15.3 ± 0.6	ELPH	1 ± 0	17.2 ± 0.7	ELPH	1 ± 0	16.3 ± 0.7
	TDAG	1 ± 0	15.7 ± 0.7	N-Gram	1 ± 0	19.1 ± 0.9	N-Gram	1 ± 0	47 ± 2
	H. N-Gram	1 ± 0	15.7 ± 0.6	H. N-Gram	1 ± 0	42 ± 1	H. N-Gram	1 ± 0	68 ± 1
	ActiveLeZi	1 ± 0	17.9 ± 0.7	PPMC	1 ± 0	61 ± 1	PPMC	1 ± 0	100 ± 2
	N-Gram	1 ± 0	19 ± 1	ActiveLeZi	1 ± 0	65 ± 1	ActiveLeZi	1 ± 0	119 ± 2
	WoLF-PHC	0.95 ± 0.01	181 ± 6	WoLF-PHC	0.89 ± 0.01	205 ± 3	WoLF-PHC	0.85 ± 0.01	210 ± 0
	ε Q-Learner	0.94 ± 0.01	37 ± 4	ε Q-Learner	0.87 ± 0.01	71 ± 5	ε Q-Learner	0.84 ± 0.01	84 ± 6
	PGA-APP	0.9 ± 0.02	144 ± 6	PGA-APP	0.87 ± 0.01	179 ± 6	PGA-APP	0.77 ± 0.01	198 ± 3
	WPL	0.63 ± 0.01	98 ± 6	WPL	0.69 ± 0.01	208 ± 2	WPL	0.76 ± 0.01	210 ± 0
	LSTM	-0.106 ± 0.001	72 ± 6	LSTM	0.006 ± 0.0002	205 ± 2	LSTM	0.012 ± 0.001	188 ± 5
N/A	KMP	1 ± 0	13.5 ± 0.5	KMP	1 ± 0	10 ± 0	KMP	1 ± 0	10 ± 0
	LZ78	0.986 ± 0.0004	84 ± 4	LZ78	0.98 ± 0.001	209.2 ± 0.7	LZ78	0.969 ± 0.002	210 ± 0
	FP	-0.335 ± 0.002	60 ± 7	FP	-0.167 ± 0.001	85 ± 6	FP	-0.11 ± 0.0008	92 ± 6

Table 4. Shows the performance of algorithm 1, using each SP-OM algorithm and fictitious play in turn as its prediction algorithm, and the reinforcement learning algorithms when played in iterated prisoner's dilemma tournaments against 20 finite automata [20]. Agent memory size is shown at the top of each part of the table. Avg Payoff is the average payoff per step over 1×10^3 testing steps ($\pm$ SEM). Avg Time is the average number of training steps required to reach the Avg Payoff ($\pm$ SEM). Position is where an agent finished in its tournament.

(a) Reinforcement learning algorithms with discount factor 0 and SP-OM algorithms with lookahead 1.

Memory Size 1				Memory Size 2			
Name	Avg Payoff	Avg Time	Position	Name	Avg Payoff	Avg Time	Position
PGA-APP	2.03 ± 0.01	30 ± 3	13	PGA-APP	2.01 ± 0.01	30 ± 4	14
ε Q-Learner	1.94 ± 0.01	30 ± 4	16	WPL	1.949 ± 0.008	20 ± 1	17
H. N-Gram	1.94 ± 0.01	30 ± 3	16	H. N-Gram	1.933 ± 0.009	20 ± 2	16
LSTM	1.939 ± 0.009	30 ± 3	16	ActiveLeZi	1.926 ± 0.009	20 ± 1	16
PPMC	1.936 ± 0.009	20 ± 1	16	LSTM	1.925 ± 0.009	20 ± 2	16
WPL	1.932 ± 0.007	20 ± 1	17	ELPH	1.922 ± 0.009	20 ± 2	16
TDAG	1.93 ± 0.01	30 ± 2	16	PPMC	1.921 ± 0.009	30 ± 2	16
ELPH	1.93 ± 0.01	30 ± 4	16	N-Gram	1.92 ± 0.009	20 ± 2	16
ActiveLeZi	1.927 ± 0.009	20 ± 1	16	WoLF-PHC	1.92 ± 0.01	30 ± 4	17
N-Gram	1.926 ± 0.009	20 ± 2	16	TDAG	1.902 ± 0.008	20 ± 2	16
WoLF-PHC	1.89 ± 0.01	20 ± 2	18	ε Q-Learner	1.822 ± 0.007	20 ± 2	18
Memory Size 3				Memory Size 4			
PGA-APP	2.02 ± 0.01	30 ± 3	14	PGA-APP	2.009 ± 0.008	20 ± 2	15
WPL	1.958 ± 0.008	20 ± 3	17	WoLF-PHC	1.978 ± 0.009	20 ± 3	16
WoLF-PHC	1.945 ± 0.009	20 ± 3	17	WPL	1.959 ± 0.007	20 ± 1	17
H. N-Gram	1.931 ± 0.009	20 ± 2	16	TDAG	1.94 ± 0.01	30 ± 3	16
N-Gram	1.931 ± 0.009	30 ± 3	16	PPMC	1.932 ± 0.009	30 ± 3	16
ELPH	1.924 ± 0.009	20 ± 2	16	ActiveLeZi	1.927 ± 0.009	20 ± 2	16
TDAG	1.92 ± 0.009	20 ± 2	16	ELPH	1.924 ± 0.009	20 ± 2	16
LSTM	1.92 ± 0.01	30 ± 3	16	N-Gram	1.92 ± 0.01	30 ± 3	16
PPMC	1.918 ± 0.009	20 ± 3	16	H. N-Gram	1.919 ± 0.009	20 ± 2	16
ActiveLeZi	1.917 ± 0.009	20 ± 2	16	LSTM	1.914 ± 0.008	30 ± 2	16
ε Q-Learner	1.773 ± 0.007	20 ± 1	18	ε Q-Learner	1.764 ± 0.007	20 ± 2	18
N/A							
KMP	1.93 ± 0.01	20 ± 2	16				
LZ78	1.927 ± 0.009	20 ± 2	16				
FP	1.922 ± 0.009	30 ± 3	16				

(b) Reinforcement learning algorithms with discount factor 0.99 and SP-OM algorithms with lookahead 2.

Memory Size 1				Memory Size 2			
Name	Avg Payoff	Avg Time	Position	Name	Avg Payoff	Avg Time	Position
ε Q-Learner	2.68 ± 0.01	180 ± 5	1	ε Q-Learner	2.74 ± 0.01	180 ± 5	1
TDAG	2.607 ± 0.008	20 ± 1	1	N-Gram	2.73 ± 0.01	30 ± 2	1
N-Gram	2.601 ± 0.007	20 ± 1	1	TDAG	2.72 ± 0.01	20 ± 1	1
ELPH	2.597 ± 0.008	30 ± 2	2	H. N-Gram	2.72 ± 0.01	40 ± 3	1
H. N-Gram	2.561 ± 0.009	30 ± 1	1	ActiveLeZi	2.7 ± 0.01	40 ± 3	1
ActiveLeZi	2.56 ± 0.01	30 ± 1	1	PPMC	2.69 ± 0.01	50 ± 4	1
PPMC	2.52 ± 0.01	30 ± 2	2	ELPH	2.51 ± 0.01	30 ± 2	2
LSTM	2.5 ± 0.01	50 ± 4	6	LSTM	2.5 ± 0.01	40 ± 3	6
WPL	2.31 ± 0.01	30 ± 4	12	WPL	2.34 ± 0.01	40 ± 4	12
PGA-APP	2.17 ± 0.02	30 ± 3	13	PGA-APP	2.18 ± 0.02	40 ± 5	13
WoLF-PHC	2.1 ± 0.02	40 ± 5	13	WoLF-PHC	2.14 ± 0.01	30 ± 3	13
Memory Size 3				Memory Size 4			
TDAG	2.74 ± 0.01	30 ± 3	1	TDAG	2.75 ± 0.01	20 ± 3	1
H. N-Gram	2.74 ± 0.01	40 ± 4	1	H. N-Gram	2.74 ± 0.01	50 ± 4	1
N-Gram	2.72 ± 0.01	40 ± 2	1	ActiveLeZi	2.72 ± 0.01	40 ± 3	1
PPMC	2.72 ± 0.01	50 ± 5	1	PPMC	2.72 ± 0.01	50 ± 4	1
ActiveLeZi	2.7 ± 0.01	40 ± 3	1	N-Gram	2.72 ± 0.01	60 ± 3	1
ε Q-Learner	2.65 ± 0.01	170 ± 5	1	ELPH	2.67 ± 0.01	140 ± 7	1
ELPH	2.54 ± 0.01	40 ± 4	2	ε Q-Learner	2.52 ± 0.01	170 ± 5	3
LSTM	2.47 ± 0.01	40 ± 2	7	LSTM	2.47 ± 0.01	40 ± 2	6
WPL	2.32 ± 0.01	30 ± 4	12	WPL	2.32 ± 0.01	30 ± 3	12
PGA-APP	2.18 ± 0.02	40 ± 4	12	PGA-APP	2.14 ± 0.01	30 ± 4	13
WoLF-PHC	2.14 ± 0.02	40 ± 4	13	WoLF-PHC	2.12 ± 0.01	30 ± 3	13
N/A							
LZ78	2.873 ± 0.008	60 ± 4	1				
KMP	2.75 ± 0.01	20 ± 2	1				
FP	1.76 ± 0.006	20 ± 3	18				

Table 4

(c) Reinforcement learning algorithms with discount factor 0.99 and SP-OM algorithms with lookahead 2 and discount factor 0.99.

Memory Size 1				Memory Size 2			
Name	Avg Payoff	Avg Time	Position	Name	Avg Payoff	Avg Time	Position
ε Q-Learner	2.68 ± 0.01	180 ± 5	1	TDAG	2.828 ± 0.009	120 ± 6	1
TDAG	2.63 ± 0.01	60 ± 4	1	N-Gram	2.817 ± 0.008	110 ± 5	1
H. N-Gram	2.62 ± 0.01	70 ± 5	1	H. N-Gram	2.817 ± 0.009	110 ± 5	1
N-Gram	2.61 ± 0.01	60 ± 4	2	ELPH	2.817 ± 0.009	120 ± 5	1
ELPH	2.6 ± 0.01	70 ± 5	3	PPMC	2.803 ± 0.008	120 ± 5	1
PPMC	2.6 ± 0.01	70 ± 4	3	ε Q-Learner	2.74 ± 0.01	180 ± 5	1
ActiveLeZi	2.54 ± 0.01	80 ± 7	4	ActiveLeZi	2.47 ± 0.01	50 ± 4	6
LSTM	2.34 ± 0.02	30 ± 3	12	LSTM	2.4 ± 0.01	30 ± 3	11
WPL	2.31 ± 0.01	30 ± 4	12	WPL	2.34 ± 0.01	40 ± 4	12
PGA-APP	2.17 ± 0.02	30 ± 3	13	PGA-APP	2.18 ± 0.02	40 ± 5	13
WoLF-PHC	2.1 ± 0.02	40 ± 5	13	WoLF-PHC	2.14 ± 0.01	30 ± 3	13
Memory Size 3				Memory Size 4			
TDAG	2.847 ± 0.009	130 ± 5	1	PPMC	2.839 ± 0.009	140 ± 5	1
N-Gram	2.831 ± 0.008	140 ± 5	1	TDAG	2.832 ± 0.009	130 ± 5	1
H. N-Gram	2.83 ± 0.009	120 ± 5	1	H. N-Gram	2.828 ± 0.009	120 ± 5	1
PPMC	2.83 ± 0.01	140 ± 5	1	N-Gram	2.826 ± 0.009	160 ± 4	1
ELPH	2.827 ± 0.009	140 ± 6	1	ELPH	2.82 ± 0.01	180 ± 4	1
ε Q-Learner	2.65 ± 0.01	170 ± 5	1	ActiveLeZi	2.59 ± 0.01	90 ± 8	2
ActiveLeZi	2.51 ± 0.01	60 ± 6	3	ε Q-Learner	2.52 ± 0.01	170 ± 5	3
LSTM	2.35 ± 0.02	40 ± 4	12	LSTM	2.36 ± 0.01	30 ± 3	11
WPL	2.32 ± 0.01	30 ± 4	12	WPL	2.32 ± 0.01	30 ± 3	12
PGA-APP	2.18 ± 0.02	40 ± 4	12	PGA-APP	2.14 ± 0.01	30 ± 4	13
WoLF-PHC	2.14 ± 0.02	40 ± 4	13	WoLF-PHC	2.12 ± 0.01	30 ± 3	13
N/A							
KMP	2.834 ± 0.008	130 ± 5	1				
LZ78	2.59 ± 0.01	90 ± 7	3				
FP	2.35 ± 0.01	30 ± 3	11				

Table 5. Shows the average payoff (fraction of goals scored/games won) per game (Avg Payoff) over 1×10^5 testing games (± SEM) by algorithm 1, using fictitious play and each SP-OM algorithm in turn as its prediction algorithm vs each reinforcement learning algorithm in Littman's soccer game.

ε Q-Learner		WoLF-PHC		WPL		PGA-APP	
Name	Avg Payoff	Name	Avg Payoff	Name	Avg Payoff	Name	Avg Payoff
PPMC	0.687 ± 0.006	PPMC	0.701 ± 0.006	PPMC	0.717 ± 0.004	PPMC	0.648 ± 0.006
LSTM	0.635 ± 0.004	LSTM	0.638 ± 0.005	H. N-Gram	0.674 ± 0.002	H. N-Gram	0.608 ± 0.003
TDAG	0.63 ± 0.004	FP	0.637 ± 0.004	N-Gram	0.665 ± 0.001	ActiveLeZi	0.599 ± 0.004
H. N-Gram	0.628 ± 0.003	N-Gram	0.614 ± 0.003	LSTM	0.659 ± 0.003	FP	0.593 ± 0.003
LZ78	0.621 ± 0.004	H. N-Gram	0.612 ± 0.003	TDAG	0.659 ± 0.002	TDAG	0.589 ± 0.004
N-Gram	0.62 ± 0.003	ActiveLeZi	0.606 ± 0.004	FP	0.655 ± 0.003	LSTM	0.585 ± 0.004
ActiveLeZi	0.618 ± 0.003	TDAG	0.606 ± 0.004	LZ78	0.653 ± 0.002	N-Gram	0.582 ± 0.003
ELPH	0.601 ± 0.004	LZ78	0.602 ± 0.004	ActiveLeZi	0.651 ± 0.002	LZ78	0.574 ± 0.003
FP	0.536 ± 0.003	ELPH	0.576 ± 0.003	ELPH	0.637 ± 0.002	ELPH	0.565 ± 0.003
KMP	0.524 ± 0.002	KMP	0.564 ± 0.003	KMP	0.62 ± 0.002	KMP	0.553 ± 0.003

7 Conclusions and Future Work

In this paper we have proposed sequence prediction opponent modelling combined with reinforcement learning and lookahead for accurately modelling and effectively responding to opponents with memory based strategies. Empirical results generally show that given enough memory and lookahead our approach has higher and faster performances against opponents of variable memory sizes compared to popular and state-of-the-art reinforcement learning algorithms. Additionally our approach outperforms these algorithms in a large zero-sum soccer game. Future work could look at deep lookahead techniques and applications to more complex domains or real-world situations.

References

1. Watkins, C.J.C.H.: Learning from delayed rewards. PhD thesis, Cambridge (1989)
2. Brown, G.: Iterative Solutions of Games by Fictitious Play. In: Activity Analysis of Production and Allocation. Wiley, New York (1951)
3. Carmel, Markovitch: Learning models of intelligent agents. In: Proc. of 13th Int. Conf. on AI, AAAI , pp. 62–67 (1996)
4. Jensen, B., Gini, S.: Non-stationary policy learning in 2-player zero sum games. In: Proc. of 20th Int. Conf. on AI, pp. 789–794 (2005)
5. Knoll, de Freitas: A machine learning perspective on predictive coding with paq. arXiv:1108.3298 (2011)
6. Treisman, Faulkner: Generation of random sequences by human subjects: Cognitive operations or psychological process? JEP: General 116, 337–355 (1987)
7. Axelrod, R.: The evolution of strategies in the iterated prisoner's dilemma. In: Genetic Algorithms and Simulated Annealing, pp. 32–41. Morgan Kaufmann (1987)
8. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: 11th Proc. of ICML, pp. 157–163. Morgan Kaufmann (1994)
9. Boytsov, Zaslavsky: Context Prediction in Pervasive Computing Systems. In: Burstein, F. (ed.) Supporting Real Time Decision-Making, pp. 35–63. Springer (2011)
10. Lempel, Ziv: Compression of individual sequences via variable-rate coding (1978)
11. Knoll, B.: Text prediction and classification using string matching (2009)
12. Moffat, A.: Implementing the ppm data compression scheme. IEEE Transactions on Communications 38, 1917–1921 (1990)
13. Gopalratnam, K., Cook, D.J.: Activelezi: An incremental parsing algorithm for sequential prediction. In: 16th Int. FLAIRS Conf., pp. 38–42 (2003)
14. Laird, P., Saul, R.: Discrete sequence prediction and its applications. Machine Learning 15, 43–68 (1994)
15. Millington, I.: Learning. In: Artificial Intelligence for Games, pp. 583–590. Morgan Kaufmann (2006)
16. Gers, F.A., Schraudolph, N.N., Schmidhuber, J.: Learning precise timing with lstm recurrent networks. JMLR 3, 115–143 (2002)
17. Bowling, M., Veloso, M.: Multiagent learning using a variable learning rate. Artificial Intelligence 136, 215–250 (2002)
18. Abdallah, S., Lesser, V.R.: Non-linear dynamics in multiagent reinforcement learning algorithms. In: AAMAS (3), pp. 1321–1324 (2008)
19. Zhang, Lesser: Multi-agent learning with policy prediction. In: AAAI (2010)
20. Piccolo, E., Squillero, G.: Adaptive opponent modelling for the iterated prisoner's dilemma. In: IEEE CEC, pp. 836–841 (2011)

Supporting Fault Tolerance in Graph-Based Multi-agent Computations

Adam Sędziwy and Leszek Kotulski

AGH University of Science and Technology,
Department of Applied Computer Science,
al. Mickiewicza 30, 30-059 Kraków, Poland
{sedziwy,kotulski}@agh.edu.pl

Abstract. Among key desirable features of a distributed system is the fault tolerance which is understood here as the ability to recover a system state after a random failure. In the article we introduce a multi-agent system (MAS) model, equipped with the mechanism increasing the reliability. The considered MAS operates on a distributed graph representation, namely on a *slashed form* of a graph, being the effective distributed graph model, which is additionally enhanced by fault-tolerance capabilities introduced for a MAS deployed on it.

Keywords: multi-agent system, fault tolerance, reliability, graph, distributed system.

1 Introduction

One of the main benefits of parallel systems application is their ability to increase the computational efficiency. To achieve this goal, one distributes fragments of centralized data to separate computing units and performs computations on them in a parallel manner. The agent system-based solutions acting according to this scheme, have to ensure the necessary cooperation and synchronization among agents. A type of the synchronization depends on a representation of a knowledge shared by agents. Most of systems are based on the service oriented cooperation and the explicit synchronization inside agents what implies problems with a verification of the system correctness. Sometimes, when a knowledge representation is formalized, an implicit synchronization rules can be introduced. Two such approaches using a graph representation of a knowledge, can be mentioned here: *complementary graph* grammars [8] and *slashed graph* grammars [12].

Distributed components may sometimes crash, so some agents (and a corresponding part of a knowledge) may be lost. Recovery of a lost knowledge, made in a distributed environment is more difficult than in a centralized one, because system state backups are made locally in an unsynchronized way. One of the possible approaches to introducing a fault-tolerance is replicating some (critical) parts of a local knowledge among several agents, so that in the case of a limited

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 397–406, 2013.

failure it may be recovered from replicated portions of a knowledge. For complementary graphs mentioned above some form of the replication was introduced in [6]. For slashed graphs no form of the replication was considered. In this paper we fill this gap by proposing the method of recovery of a knowledge represented by a slashed graph when a single agent was crashed.

The paper is organized as follows. In the next section the brief overview of related works is made. The concept of a slashed representation and related issues are included in Section 3. In Section 4 we introduce a multi-agent system capable of recovering a lost knowledge: its architecture, details of the recovery mechanism and the case study. The paper conclusions are contained in Section 5.

2 Related Works

Localizing and removing failures of a multi-agent system is a complex task due to the distributed nature of an environment. Moreover a single fault may propagate across the entire system through agents interactions. To prevent a multi-agent system failures or to handle such events, various methods are used.

Ensuring a reliability of multi-agent systems is an object of the intensive research for several years [2,10,11]. Among the most intuitive and popular approaches is replicating critical agents, extended sometimes by adaptation and prediction capabilities [1,3,4]. This method increase, however, system load and complexity thus it has to be either applied according to certain policies or accompanied by supplementary techniques decreasing a system load, e.g., a transparent agent replication [3].

For a multi-agent system deployed on a distributed graph structure, namely, on *replicated complementary graphs*, the fault-tolerance is supported by the partial replication of a graph structure [6,7]. Although a recovery mechanism is not introduced formally in this case, a basic knowledge about a system, which is represented by graphs, can be maintained inside replicated nodes. Thus it is not lost when a single agent crashes. The agent GRADIS framework (GRAph DIStributed) based on this paradigm was presented in [8].

Another, more theoretical research concerns the possibility of achieving a global task after a system failure [5]. These considerations are made for top-down design approach where a global task is decomposed into several subtasks to be achieved in parallel. A multi-agent system considered in this paper works in such manner

The approach to a system reliability, presented in this paper focuses on a system recovery rather than on avoiding system failures or bypassing them with the help of redundancy or the work redistribution.

3 Slashed Graphs Model

As mentioned previously, the centralized problem formulation is often replaced with a distributed one: a global goal is achieved by solving multiple subtasks. Hence a centralized, graph model has to be replaced with a distributed graph representation of a system.

3.1 Slashed Form of a Graph

The decomposition of a centralized graph is the step preparing an environment for a multi-agent system deployment.

Preserving the system description consistency after a graph decomposition requires that agents share some information related to nodes or/and edges located in border areas. It may however imply an overhead in agents coordination for operations performed on such a shared elements. Thus a decomposition method influences a performance of a multi-agent system.

The representation which enables minimizing the coordination complexity is the *slashed form* of a graph. It was introduced in [12] to reduce coupling among subgraphs (obtained from decomposition of a centralized graph) and thereby to simplify operations performed by a multi-agent system.

Figure 1(a) shows the exemplary graph G, being a centralized representation of a system. The dotted lines mark boundaries of the further decomposition. Figure 1(b) contains the **slashed** (i.e., decomposed) **form** of the graph G, denoted as $\not{G}$. Subgraphs of $\not{G}$ (in the considered case G_1, G_2, G_3) are referred to as **slashed components** of G. Newly appeared vertices in $\not{G}$, marked as squares, obtained in a result of edges slashing, are referred to as **dummy nodes**. All vertices marked as circles in Fig.1(b), existing previously in the graph G, are called **core nodes**. Let E_i be a set of edges of a slashed component G_i, then an edge $e \in E_i$ is called a **border** one when it is incident to a dummy node. If both endpoints of e are core vertices then we call it a **core** edge. It is assumed that to each a core edge the attribute EP (EndPoints) is assigned. EP contains identifiers of vertices incident to this edge. Analogously, to each a border edge we ascribe EP attribute with identifiers of endpoints of an underlying core edge.

Example. For the slashed form shown in Fig. 1 the underlying edge for border edges $((1,2),(-1,3))$ and $((-1,3),(2,4))$ is the core one: $\{v_5, v_6\}$.

The following indexing convention is used for slashed components (see Fig. 1a). A core node index has the form (i,k) where i is an unique, within $\not{G}$, identifier of a slashed component G_i and k is an unique, within G_i, index of this node. A dummy node index has the form $(-1,k)_r$ where k is a globally unique identifier of a node. Additionally, a subscript r denotes a reference to a slashed component (or its maintaining agent) hosting a replica of a given dummy node. Using such a subscript allows for immediate localization of a replica. To simplify the notation subscripts will be neglected within the text, unless needed. Note that a dummy vertex and its replica share a common index and differ in reference subscripts only: $(-1,k)_{r_1}, (-1,k)_{r_2}$.

3.2 Incorporate Procedure in Slashed Components Environment

We assume that an agent either has a sufficient knowledge (represented by its slashed component) or imports it form other agent(s). Incorporate procedure which allows for importing a part of slashed component (representing a knowledge) is the most frequently performed operation in an agent system being considered (more precisely, it is performed by *computing agents* (CA) defined in

Section 4.1). For this reason its efficiency impacts strongly on the overall system performance. A slashed component's border consisting of all its dummy nodes, may be shifted by calling Incorporate on a given border edge e_v incident to a dummy vertex v. It is accomplished by matching v and its replica v' being an endpoint of a border edge $e_{v'}$ in another component and recovering an underlying edge from e_v and $e_{v'}$. Thus exactly one core node and possibly some dummy ones are attached to a given component. Figure 1b changes in $\mathcal{G}$ implied by two subsequent incorporations: of the border edges $e_{(-1,1)}$ and $e_{(-1,3)}$, into G_1 which will be called an initiating component. Let us consider the first incorporation in detail.

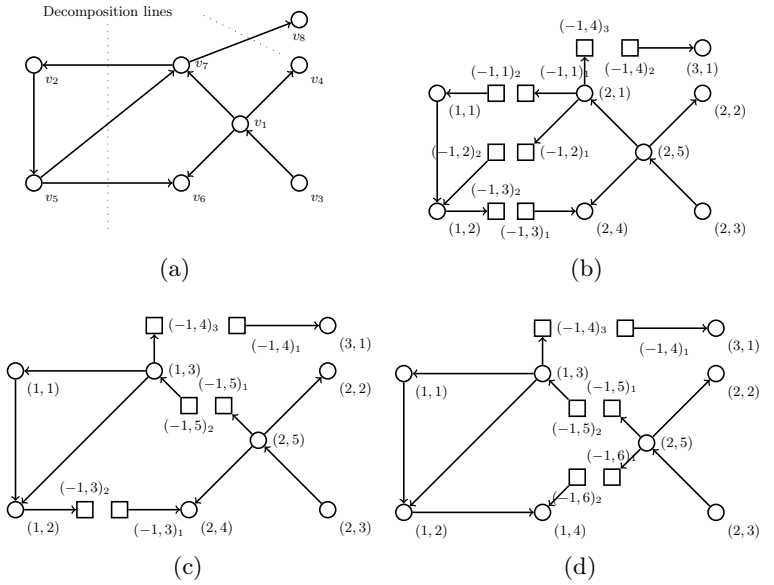


Fig. 1. (a) The centralized graph G (b) $\mathcal{G} = \{G_1, G_2, G_3\}$ (c) $\mathcal{G}$ after incorporating $e_{(-1,1)}$ to G_1 (d) $\mathcal{G}$ after incorporating $e_{(-1,3)}$ to G_1

The operation Incorporate(G_i, e_v) for G_1 and the border edge e_v incident to the dummy node $v = (-1,1)_2$ consists of following steps:

1. Get the replica of v, namely $v' = (-1,1)_1$, localized in G_2 (as referenced by the subscript of the index of v).
2. Move the core node u indexed by $(2,1)$, neighboring v', to G_1 and reindex u to $(1,3)$ (for the compliance with the indexation in G_1).
3. Move to G_1 all dummy nodes neighboring u in G_2, together with incident border edges. Let us assume WLOG that a border edge $e_d = (u, m, d)$. Two scenarios are possible. In the first one a dummy vertex d (in the example d indexed with $(-1,2)$) has a replica d' in an initiating graph. In that case d, d' are removed and a corresponding slashed edge e is recovered (in Fig.1

$e = ((1,3), m_e, (1,2)))$. In the second scenario d does not match any dummy vertex in an initiating graph and it is attached together with e_d to an initiating graph. Note that e_d may not be split because an edge belonging to a slashed component may be incident to at last one dummy node.

4. Split all edges connecting u with other core nodes. Resultant dummy vertices are attached to the initiating graph together with incident edges. The edge $e = ((2,5), m_e, (2,1))$ shown in Figure 1 has been split, resultant $e_{(-1,5)} = ((-1,5), m, (1,3))$ has been attached to G_1 (note that u was reindexed previously) and $e'_{(-1,5)} = ((2,5), m', (-1,5))$ remains in G_2.

Incorporate(G_0, e_v) procedure execution has one phase only. Let us assume that v' is a replica of v and c is a core node being the neighbor of v'. To perform Incorporate(G_0, e_v) an initiator agent (say A_0) sends a `commit` request to a responder agent (say A_1) to gather c vertex and its neighborhood (which may contain newly created dummy nodes). A_1 may either supply requested nodes/edges or reject a request (a response is `abort`) if one or more of requested nodes/edges are already locked by some other initiator. If A_0 was replied with a requested subgraph then it sends `update` requests to agents maintaining replicas of all dummy vertices which just have been attached to G_0. A_0 requests to update references for replicas of those vertices. Note that references to those agents are known to A_0 as they are included in dummy nodes' indices. It is assumed that each an agent keeps the information where his particular dummy nodes were moved to. Hence that if an agent being a recipient of an `update` message has not a dummy vertex replica u' to be updated, because u' node was previously moved to some other slashed component G_k, then it forwards `update` request to the corresponding agent A_k and informs A_0 that a replica location changed. In the case when a response from A_1 is `abort`, A_0 repeats the operation with a random delay.

The Incorporate protocol described above is more efficient compared to its form used in the case RCG representations which acts according to the 2PC protocol semantics [9].

Figure 1c demonstrates an index update: after incorporating $e_{(-1,1)}$ by G_1 the index $(-1,4)_2$ in slashed component G_3 has changed to $(-1,4)_1$.

A border edge e being incorporated to G_j from G_i is also referred to as *handed over* from G_i to G_j.

3.3 Incorporation Tracking

Let A_i be a computing agent managing a slashed component G_i. After incorporating a border edge, A_i submits two timestamped notifications containing an event description, to an agent called *dispatcher*. One of notifications is sent by A_i on its own behalf, the second one on behalf of the other agent participating in an incorporation. Tracking such events enables introducing a system recovery mechanism described further.

We assume that a computing agent's notification record (NR) has the following structure:

$$\text{NR} = \{\texttt{timeStamp}, \texttt{incorpGUID}, \texttt{opType}, \texttt{cooperatingAgent}, \texttt{cn}, \texttt{ncn}\},$$

where `timeStamp` is an identifier of an incorporation time; `incorpGUID` is a globally unique identifier of an incorporation; `opType` specifies an operation type: -1 when a NR is sent on behalf of an agent which handed over a given border edge e or $+1$ when e was incorporated to G_i by A_i; `cooperatingAgent` is a reference to an agent participating in incorporation either as a responder (if `opType` $= +1$) or as an initiator (when `opType` $= -1$); `cn` is an identifier of a core node which was either added to or handed over from G_i in result of an incorporation; when `opType` $= +1$ then `ncn` is a list identifiers of core nodes being a neighbors of `cn` in G_i; when `opType` $= -1$ then `ncn` is a list of identifiers of core nodes which were neighbors of `cn` in G_k (assuming that `cn` was moved from to G_i from G_k).

Remark: A reference `cn` and references in `ncn` are globally unique identifiers of corresponding core nodes. It is assumed that such an identifier is stored in a dedicated core node's attribute.

A dispatcher agent holds a database (denoted as REG in Figure 3) of separate registries of notification records for each a CA (in Figure 3 denoted as RNR_1, RNR_2,...). NRs are stored in a submission order, i.e., they are ordered by time stamps. Additionally, a registry assigned to a given CA holds a snapshot of the corresponding slashed component. Snapshots are submitted by a computing agent periodically. A snapshot submission causes clearing corresponding registry and subsequent notifications are relative to the state given by a current snapshot. Maintaining a snapshot resolves the problem of a limited registry size.

Note that the registry structure introduced above enables tracking changes in a system state, in particular changes in actual locations of core nodes.

Example. Let us assume that for $\not{G}$ shown in Figure 1 we have three computing agents A_1, A_2, A_3 ascribed to G_1, G_2, G_3 respectively. Additionally let Fig.1b represents the initial state of $\not{G}$. The initial state of particular RNRs is shown in Tables 2(a)-(c). Corresponding snapshots of slashed components are equal to G_1, G_2, G_3 respectively. Due to two incorporations shown in Figure 1, two registries of notification records change to the form presented in Tables 2 (d),(e).

4 System Recovery Supported by MAS

In this section we focus on ensuring the fault tolerance of a multi-agent system. In particular the case study of the system recovery will be considered.

4.1 Architecture of MAS

In the considered multi-agent system following types of agents are defined (see Figure 3).

(a) RNR_1

time	guid	opType	agent	cn	ncn
-	-	-	-	-	-

(b) RNR_2

time	guid	opType	agent	cn	ncn
-	-	-	-	-	-

(c) RNR_3

time	guid	opType	agent	cn	ncn
-	-	-	-	-	-

(d) RNR_1

time	guid	opType	agent	cn	ncn
1	id_1	+1	A_2	v_7	v_2, v_5
2	id_2	+1	A_2	v_6	v_5

(e) RNR_2

time	guid	opType	agent	cn	ncn
1	id_1	-1	A_1	v_7	v_1
2	id_2	-1	A_1	v_6	v_1

(f) RNR_3

time	guid	opType	agent	cn	ncn
-		-	-	-	-

Fig. 2. The initial state of the REG database (left column of tables) and its content after incorporations shown in Fig.1

- Dispatcher Agent (DA) which primarily decomposes a main graph into slashed components, creates *computing agents* and deploys them into produced subgraphs, and creates *recovery agent* if needed. It also maintains the registry of notification records and tracks heartbeat messages sent by computing agents. If such a message is missed then a recovery action is triggered. Only one DA is present in a multi-agent system.
- Computing Agent (CA) performs computational tasks and sends obtained results to a DA. A computing agent may incorporate border edges from neighboring slashed components. Then, after completion, it sends relevant NRs to a dispatcher agent. Another, background activity of a CA, is sending periodically heartbeat messages to a dispatcher agent, to confirm CA's alive status.

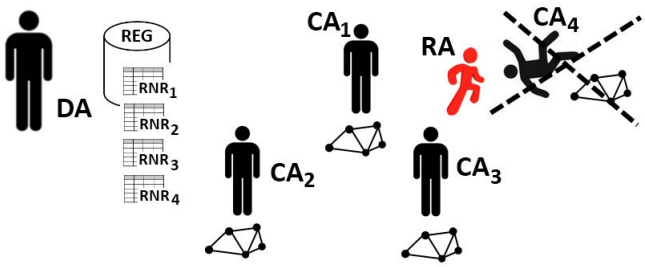


Fig. 3. Multi-agent system architecture

4.2 MAS Reliability: Recovery Agent

A potential MAS failures may result from following possible reasons:

- an agent's platform (e.g., JADE container) breakdown caused by software or hardware errors,
- a loss of connection with a computing agent.

If a failure occurs then a Recovery Agent (RA) is delegated by a dispatcher agent to restore a lost slashed component and to recreate computing agent assigned to it. Dispatcher Agent may create multiple recovery agents dependently on actual needs.

Algorithm 1. Recover(i, REG)

Input: i - identifying number of a slashed component to be recovered,
REG - set of registries containing notification records of particular agents
Result: recovered slashed component G_i

```
1  begin
2      RNR_i ← REG(i);                    /* Get the registry for i-th agent */
3      G_i ← RNR_i.getSnapshot();          /* Get the latest snapshot of G_i */
4      V_i^+ ← set of core nodes specified by NRs ∈ RNR_i, having opType = +1;
5      V_i^- ← set of core nodes specified by NRs ∈ RNR_i, having opType = −1;
6      V_i^⋆ ← V(G_i) ∪ V_i^+ \ V_i^-;
7      foreach v ∈ V_i^- \ V_i^+ do
8          Remove all edges incident to v;
9          foreach core node w neighboring v in G_i do
10             CA ← a computing agent's reference retrieved from the NR relevant
               to v;
11             Query CA for the dummy node d incident to a border edge e such
               that EP(e)={w, v};
12             d ← Get response from CA;
13             Add an edge {w, d} to E(G_i);
14     foreach v ∈ V_i^+ \ V_i^- do
15         Recover all core edges incident to v in G_i;
16         L_c ← set of core nodes neighboring v and non-belonging to V_i^⋆;
17         L_A ← set of computing agents maintaing slashed components
           containing nodes belonging to L_c;
18         foreach A ∈ L_A do
19             Query CA for the dummy node d incident to a border edge e such
               that v ∈EP(e);
20             d ← Get response from CA;
21             Add an edge {v, d} to E(G_i);
22     V(G_i) ← V_i^⋆;
```

An RA aims at ensuring a multi-agent system fault-tolerance. It is accomplished by using a list of latest snapshots of slashed components and a registry of notifications in cooperation with corresponding CAs. The first step towards recovering a lost slashed component is recreating it in its last known state, i.e., in a form obtained from an actual snapshot and not affected by subsequent incorporations. Next, particular nodes (dummy and/or core ones) have to be either removed or recovered, dependently on a further history of a restored slashed component. The pseudocode presenting this operation in detail is shown in Algorithm 1.

Core edges (line15), a list of core nodes (line 16) and a list of agents (line 17) are retrieved by a recovery agent from data stored in REG database.

Let us note that a recovery agent doesn't require a knowledge on a centralized graph G. Moreover, such a knowledge gets useless when G is mutable, e.g., when a multi-agent system makes some distributed transformations on $\mathcal{G}$.

4.3 System Recovery – Case Study

Let us consider two scenarios occurring in the environment of slashed components shown in Figure 1d.

In the first scenario we assume that the agent CA_1 and its slashed component, G_1, are lost. In this case the recovery agent starts restoring G_1 taking its snapshot visible in Figure 1b. Next, the RA reads subsequent records from RNR_1 and adds vertices and edges to G_1 accordingly: (i) nodes v_6 and v_7 from G (reindexing them respectively to $(1,4)$ and $(1,3)$), as found in records 1 and 2, (ii) core edges: $((1,3),(1,1))$, $((1,2),(1,3))$, $((1,2),(1,4))$. Next the RA queries CA_2 for dummy nodes which replicas have to be added to G_1: $(-1,4),(-1,5),(-1,6)$. After obtaining those data from CA_2 and CA_3, missing border edges are added to G_1: $((1,3),(-1,4))$, $((1,3),(-1,5))$ and $((1,4),(-1,6))$. After completing these steps RA recreates CA_1, ascribes it to G_1 and terminates.

In the second scenario the agent CA_2 is lost. The recovery process starts form the snapshot G_2 as seen in Figure 1b. In this case, however, the registry (RNR_2) records for CA_2 imply removing core nodes $(2,1)$, $(2,4)$ and all incident edges. Having a reference to an agent which initiated both incorporations (namely CA_1) RA requests it to provide data on dummy vertices neighboring currently the core nodes being incorporated and shared with G_2. Thus replicas of $(-1,5)$ and $(-1,6)$ are created and added to G_2 together with incident border edges. As previously, after completing G_2 recovery, RA recreates CA_2, ascribes it to G_2 and terminates.

5 Conclusions

The reliability is one of the most important properties of a multi-agent system, especially in computational tasks. The approach commonly used to achieve the reliability is introducing some forms of a redundancy e.g. knowledge replication.

Slashed graphs provide both an efficient method of distributing and maintaining a graph-represented knowledge, and the environment for multi-agent system deployment. An agent failure however causes problems with a knowledge integrity, even if local slashed components are periodically backed up in some safe space. This undesired effect results from the possibility of incorporations which impact slashed components and thereby a local knowledge state. In the paper we proposed the recovery technique for solving this problem, based on a local knowledge backups (when a knowledge is represented by means of graphs) and incorporations tracking. From the formal point of view the introduced recovery algorithm has $O(n^2)$ computational complexity, where n is a number of nodes,

but usually V_i^+ and V_i^- sets have a few elements only so it makes the algorithm fast enough to be used in practice.

The proposed recovery method is also applicable in the case when a global knowledge state is not static, i.e., when a graph grammar's productions are applied to a graph. The presented approach will be applied in computations supporting a lighting design in Green AGH Campus smart grid project.

References

1. Almeida, A.L., Aknine, S., Briot, J.-P., Malenfant, J.: Plan-based replication for fault-tolerant multi-agent systems. In: 20th International Parallel and Distributed Processing Symposium, IPDPS 2006 (April 2006)
2. Bora, S., Dikenelli, O.: Implementing a multi-agent organization that changes its fault tolerance policy at run-time. In: Dikenelli, O., Gleizes, M.-P., Ricci, A. (eds.) ESAW 2005. LNCS (LNAI), vol. 3963, pp. 153–167. Springer, Heidelberg (2006)
3. Fedoruk, A., Deters, R.: Improving fault-tolerance by replicating agents. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2, AAMAS 2002, ACM, New York (2002)
4. Guessoum, Z., Faci, N., Briot, J.-P.: Adaptive replication of large-scale multi-agent systems – towards a fault-tolerant multi-agent platform. In: Garcia, A., Choren, R., Lucena, C., Giorgini, P., Holvoet, T., Romanovsky, A. (eds.) SELMAS 2005. LNCS, vol. 3914, pp. 238–253. Springer, Heidelberg (2006)
5. Karimadini, M., Lin, H.: Task decomposability under event failures for two cooperative agents. CoRR, abs/1101.2003 (2011)
6. Kotulski, L.: On the control complementary graph replication. In: Mazurkiewicz, J., et al. (eds.) Models and Methodology of System Dependability. Monographs of System Dependability, vol. 1, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław
7. Kotulski, L.: Distributed graphs transformed by multiagent system. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 1234–1242. Springer, Heidelberg (2008)
8. Kotulski, L.: GRADIS – Multiagent Environment Supporting Distributed Graph Transformations. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part III. LNCS, vol. 5103, pp. 644–653. Springer, Heidelberg (2008)
9. Kotulski, L., Sędziwy, A.: Parallel graph transformations supported by replicated complementary graphs. In: Dobnikar, A., Lotrič, U., Šter, B. (eds.) ICANNGA 2011, Part II. LNCS, vol. 6594, pp. 254–264. Springer, Heidelberg (2011)
10. Potiron, K., Taillibert, P., El Fallah Seghrouchni, A.: A step towards fault tolerance for multi-agent systems. In: Dastani, M., El Fallah Seghrouchni, A., Leite, J., Torroni, P. (eds.) LADS 2007. LNCS (LNAI), vol. 5118, pp. 156–172. Springer, Heidelberg (2008)
11. Rudnianski, M., Bestougeff, H.: Multi-agent systems reliability, fuzziness, and deterrence. In: Hinchey, M.G., Rash, J.L., Truszkowski, W.F., Rouff, C.A. (eds.) FAABS 2004. LNCS (LNAI), vol. 3228, pp. 41–56. Springer, Heidelberg (2004)
12. Sędziwy, A.: Effective graph representation for agent-based distributed computing. In: Jezic, G., Kusek, M., Nguyen, N.-T., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2012. LNCS, vol. 7327, pp. 638–647. Springer, Heidelberg (2012)

moviQuest-MAS: An Intelligent Platform for Ubiquitous Social Networking Business

Ramon Soto[1], Alexandro Soto[2], and Juan Camalich[3]

[1] University of Sonora, Department of Accounting, Hermosillo, Sonora, México
rsotoc@moviquest.com
[2] University of Sonora, Department of Mathematics, Hermosillo, Sonora, México
alexsf@moviquest.com
[3] moviQuest, S.A. de C.V., Department of Engineering, Hermosillo, Sonora, México
juanc@moviquest.com

Abstract. New communications technologies have revolutionized human interaction. One of the main features of this new order is an explosion of relationships a person can establish and the issues that she can start. However, keeping control of the course of these relationships has become a very complex issue. This paper presents a generic platform for implementing social networks which aim to establish serious relationships between people based on their membership to affinity groups. The proposed social network platform uses an artificial intelligence approach which has been designated as Ubiquitous Artificial Intelligence. The basis of this approach is a multi-agent architecture that integrates a variety of pattern recognition algorithms to determine the profiles of social network use along with mobile expert systems that assist the user in their social network communication from the mobile. This approach also uses a specific ontology of the social network that simplifies communication and makes it safer.

Keywords: Social networks, Mobile computing, Intelligent computing, Multiagent technologies.

1 Introduction

The cyberspace has become the place where people spend much of their time in the current era. People increasingly feel more comfortable managing their relationships through digital media than in a face to face way. The rapid development of mobile technology come to accelerate this process of virtualization of human life. However, in many cases social networking sites have become places for gossip with little real benefits and several risks for the user.

In this paper we introduce moviQuest-MAS, a generic software platform for implementing serious social networks based on affinity users groups. This platform constitutes a software ecosystem consisting of a web application, a desktop application and a family of mobile applications, which together offer a high degree of accessibility and security. An artificial intelligence model called Ubiquitous Artificial Intelligence is introduced, which allows to automate the key

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 407–418, 2013.

activities in the social network. The ontology used for communicating software agents optimize the communication between them, reducing the communication costs between the social network and the mobile devices. At the same time, this ontology provides a basis for secure messaging.

2 Brief Literature Review

Social networks are one of the greatest technological phenomena of recent years, featured as the second most important technology of the past decade, just behind the revolution caused by smartphones [12]. Thus, not surprisingly, these themes are strongly calling the attention of both academics and industry. Due to space constraints, we will review just a little of the recently published related work.

Maamar et. al. [9] and Al Falahi et. al. [6] stand out the rich possibilities offered by social networking as a business platform. Yüksel et. al. have emphasized the use of interest groups within social networks to help maintain safety. Canali et. al. [5] and Cameron et. al. [4], among others authors, have discuss different methods to automatically identify diverse characteristics of the social network users. Bodriagov and Buchegger [3], Lan et. al. [8] and other research teams have presented interesting ideas on how to improve the privacy of user data in social networks. Bergenti et. al. [1] and Mavridis [10], in turn, have highlighted the benefits of introducing software agents in social networks. In the latter case, the results of introducing Sarah, a physical mobile robot, as user of the social network Facebook are presented.

3 moviQuest-MAS Software Ecosystem

moviQuest-MAS consists of a set of software solutions that conforms what can be called a software ecosystem [2]. The three main components in moviQuest-MAS are the following:

- **Control center software**. This component is a desktop application designed to run on the control room of the organization, giving the system manager and operators complete control over system resources and keeping sensitive data away from the risks inherent to a Web application. Its main functions are to manage the content offered to users (news, documents, applications, advertising, etc.) and concentrate the information of interest to the organization owning the social network such as user profiles, usage patterns, characterization of leadership, etc.).
- **Social network engine**. This application provides access to the virtual community through a standard web browser, including the typical activities or features in a social network such as microblogging, groups creation, user registration, posting of events, news and documents, etc.
- **Mobile client**. This component is a mobile application that allows communications with the organization from a mobile phones (both feature phones

and most smartphones) using encrypted SMS messages. Through this application, the client receives and sends information to and from the organization and to the social network.

The modules of software in the moviQuest-MAS ecosystem offers three desirable features in any system of open information: 1) the benefits of access through a Web interface, a feature essential for a social network platform, 2) a high level of security, somewhat easier to secure by desktop applications than by web applications, and 3) a wide remote access to the system based on cheap phones and a channel of communication with high availability worldwide.

4 moviQuest-MAS Functionality

moviQuest-MAS provides a platform for implementing social networks based on affinity groups. Interaction of individuals in such a social network is restricted to those people with which the user has any affinities. This interaction scheme mimics the interaction on off-line social networks: All social interaction in the real world takes place within a social structure based on family, community, hobbies, religious or political affiliation, work, friendship, or any other affinity relationship.

Each affinity group in a moviQuest-MAS social network has a *group manager* who exerts formal leadership role. Each group may be public or private, a feature that is defined at the time of its creation and can be modified at any time by the group manager. Any group member can invite other people to join the group. Also, anyone on the social network can apply to become member of all groups that interest him.

User profiles in a moviQuest-MAS social network are managed as private data. Only users belonging to a group can see the list of group members and some of the corresponding profile data. Some data from group manager profile are always available in the group description for any registered user.

The social network also provides a process for transferring formal leadership, by a series of mechanisms defined by the organization owning the network. Some mechanisms available include the transfer, substitution and referendum. In the latter case, the transfer is done through a process of claim and vote. The referendum can also be employed to expel a member whose participation is considered harmful to the group. This moviQuest-MAS feature gives the affinity groups the flexibility to self-organize and evolve.

The main functions available on moviQuest-MAS are presented in the following sections:

4.1 moviQuest Microblogging

Microblogging is a service that enables short messages publishing in a dedicated website. This is the main form of service and interaction within a social network. Microblogging messages in a moviQuest-MAS social network can be of three types:

- Public messages: These kind of messages are targeted for all members of all the groups to which the author of the message belongs.
- Group messages: These messages are used to share information with all the users in specific groups, to which the author of the message must belong.
- Private messages: These are messages sent to a user or group of users within the groups to which the author belongs, regardless of whether all they belong to the same groups.

An important feature of moviQuest-MAS social network is their ability to start conversations from mobile phones. The messages from a mobile device are sent through a special application that allows user to specify the user or group to whom the message is addressed. The application also allows the user to send copy of each message via SMS to the addressees. Thus, the user benefits from sending or receiveing notifications from their contacts to and from mobile devices, while keeping the Web blackboard to review, analyze and continue the conversations generated from the cell phone.

4.2 Real Time Surveys

One of the most important functionalities of moviQuest-MAS is a framework for the design and implementation of adaptive surveys via mobile phones. This tool allows to develop complex questionnaires to be sent to cell phones of registered users using one to three concatenated SMS messages. The questionnaires are received into a special port on the phone by the same software used to send messages to the social network. The answers are sent to the server using also a short message. Both the questionnaire and the response are encoded using an ontology specific to the organization and then encrypted. This coding significantly reduces the size of the packets of information and together with the encryption process guarantees a high level information. The SMS channel usage makes it unnecessary for the receiving phone to have Internet access and neither involves the typical cost of web access from a mobile device. The survey responses are received on the moviQuest-MAS server where they can be analyzed as they are being received. These results can be published directly to a section on the website of the social network. The results can also be sent via SMS to predefined users.

4.3 Spontaneous Evaluations

The software application in the mobile phone and the use of SMS messages for communicating with the social network server opens diverse possibilities to interact with the user. One of these options is the collection of spontaneous evaluations, that is to say, evaluations that the user is interested in carrying out about the services it obtains from the organization, without their being requested. A citizen, for example, may be interested in expressing their opinion about public services, a customer may want to evaluate the quality of service of a company and a student may want to evaluate their teachers, without waiting for a survey from the organization. This information is very important to the organization as an indicator of its public image in the community and about the client satisfaction.

4.4 Active Media Monitoring

moviQuest-MAS provides a means for real-time monitoring and quickly processing of several electronic sources for relevant information to the group. The notes obtained are automatically presented on a screen where the system operator analyzes and selects those suitable for publishing to the social network. These notes appear in a special section in the Web site.

4.5 Mobile Marketing

To have a permanent connection with a group of users with well-identified affinities opens up exciting possibilities for marketing. The user profile data, together with their membership in one or more groups and other elements of its activities enable the organization that owns the social network to create accurate profiles of their customers. With this information in hand, the organization can design marketing strategies directed to well-defined target populations.

5 Ubiquitous Artificial Intelligence

Intelligence is a difficult concept to define and has led to many debates (see for example [11] for a broad discussion about the definition of intelligence). There is, however, a broad consensus on abandoning the traditional definition of intelligence as the human ability to handle abstract concepts and instead define it as the ability of living beings to adapt to their environment. The new way of understanding intelligence is based on concepts such as learning, pattern recognition, reasoning under ambiguity, self-organization and collaboration.

moviQuest-MAS uses a Soft Computing approach, named as Ubiquitous Artificial Intelligence, to provide the organization that owns the social network and its users with a set of intelligent tools that enhance their interaction in the virtual community. This approach is based on three main components:

- A multiagent architecture in which coexist human agents (users and managers of the social network) with different classes of software agents.
- The *mobile smart agent*, a light expert system that can run on most feature phones and smartphones.
- MQDM: A method for fuzzy group decision making in structured social networks.

5.1 MAS Architecture

The multiagent architecture of moviQuest-MAS (see figure 1) is based on the AGR (Agent-Group-Role) model proposed in [7]. In this organization centered multi-agent system model, the interaction between agents is characterized by the roles they play into the groups they belong to. AGR multiagent model places no constraints upon the model of each agent and their individual capabilities. This multi-agent approach clearly reflects the philosophy of moviQuest-MAS whereby

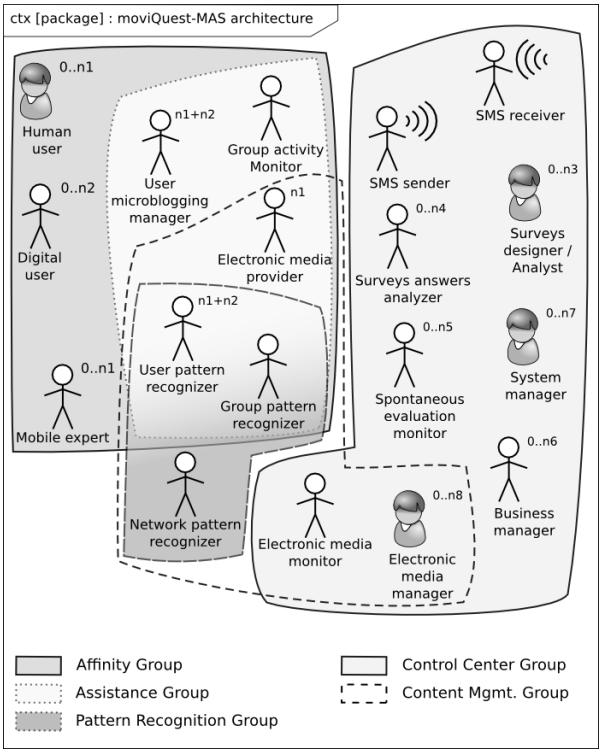


Fig. 1. moviQuest-MAS multiagent architecture

the interaction between individuals belonging to the social network is focused on the organizational structure.

moviQuest-MAS can be described as:

$$\Xi_\alpha = (\mathbf{A}, \mathbf{G}, \mathbf{R}, \mathcal{K}_\alpha) \tag{1}$$

where

- $\mathbf{A}$ is the set of agents.
- $\mathbf{G}$ is the set of groups.
- $\mathbf{R}$ is the set of roles an agent can play.
- $\mathcal{K}_\alpha$ is the knowledge base used by agents for a specific domain α.

Agents. Agents in moviQuest-MAS can be both human agents and software agents. Human agents fall into two main classes: users and managers, each with the potential to play diverse roles within a moviQuest-MAS business network. Software agents, on the other hand, are designed to automate virtually every task in the system. A single software agent can perform different sets of functions, which are represented by specific roles of agents. Thus, the functionality of the system can be expanded simply by adding new agents with a set of roles to meet the new needs.

Groups. The activities of any agent, whether software or human, is determined by the groups they belong and by the function of these groups within the system. The main groups in a moviQuest-MAS system are the following:

- *Affinity Group.* These are the basic groups for social networking in moviQues-MAS.
- *Assistance Group.* These groups, one for each affinity group, are formed by software and human agents that provide personalized assistance to the user.
- *Pattern Recognition Group.* This group is intended to recognize user patterns within the different affinity groups. It is composed by software agents from each affinity group, along with global agents that integrate information across the social network and human staff dedicated to analyze the data obtained by software agents.
- *Control Center Group.* This group is responsible for managing the entire system. The main agents in this group are system administrators along to software agents which help to automate common tasks.
- *Content Management Group.* This group has the function of selecting electronic content which are expected to be of interest to the users in each affinity group, using the information gathered by the pattern recognition group.

Roles. The main roles a human agent can play within a moviQuest-MAS group are:

- *Human user.* It is the generic role played by a person for social networking.
- *Group manager.* The main functions of this role are to respond to affiliation requests, to authorize the publication of events in the group and deal with complaints about users misconduct.
- *System manager.* It is responsible for maintaining the proper functioning of the social network platform.
- *Electronic media manager.* This role is responsible for selecting and publishing information that may be relevant to the diverse affinity groups.
- *Surveys designer.* This role is responsible for designing the surveys that will be sent via mobile clients to registered users in a given group.
- *Surveys analyst.* This role includes the tasks required to analyze and publish the results of the surveys.

On the other hand, software agents can play the following roles:

- *Mobile expert.* This role includes a set of tasks that allow human users to participate in the social network through a mobile device.
- *Digital user.* This role allows organisations to interact automatedly with human users in the social network through softbots.
- *Group manager.* Although this role is intended to be executed by a human user, in some circumstances it may be convenient to assign this activity to a software agent. For example, when the group membership is decided by well-defined facts (age, place of residence, university degree, etc.).

- *SMS receiver*. This role is responsible for receiving and managing SMS messages arriving from mobile clients.
- *SMS sender*. This role is responsible for managing the sending of SMS messages with information from the social network to mobile clients.
- *Surveys answers analyzer*. This role comprises the tasks required for the preprocessing of the survey answers.
- *Spontaneous evaluations monitor*. This role is intended to manage the spontaneous evaluations from the mobile clients.
- *Electronic media monitor*. This role is responsible for looking for information in the Web that may be of interest to the different affinity groups.
- *Electronic media provider*. This role is responsible for managing the information obtained from the Web to publish it to the affinity groups blackboards.
- *User microblogging manager*. This role includes customer support functions for microblogging activities, such as automatic responses, suggestions for new affinity groups, reminders of events, etc.
- *User pattern recognizer*. This role is intended to detect the user activity patterns in order to make him intelligent suggestions about events, new information, pending discussions, etc.
- *Group pattern recognizer*. The main task in this role is to integrate the information collected about the activity patterns of all users in an affinity group in order of detect specific groups needs an interests.
- *Network pattern recognizer*. This role has as its primary function to collect information from each of the affinity groups to identify and to integrate collective needs and concerns of social network users.
- *Business manager*. This is a generic role can range from suggesting new relationships between social network users to implement security algorithms required for business transactions.

Knowledge Base. The knowledge base $\mathcal{K}_\alpha$ contains the ontology along with the set of rules used by the expert system in its inference process. The knowledge base is basically a tree grammar. A specific questionnaire is represented by a tree with multiple branching generated by this grammar (figure 2). Such a tree can contain four types of nodes: explicit rule nodes, blank nodes, concept nodes and terminal nodes (leaves). Explicit rule nodes have associated a question to be presented to the respondent. The blank nodes correspond to non-explicit rules, ie rules that do not have an associated question to submit to the respondent. The rule, in this case is triggered by selecting an option to the question asked by the parent node. Concept nodes correspond to features that validate a rule and are presented as a question to the respondent. The terminal nodes represent the final answer on a path and triggers by selecting an option in the parent node. The result of the questionnaire is given by all terminal nodes that were fired.

The use of $\mathcal{L}_\alpha$ allows to send a large set of questions (up to 60 questions or so) using no more than three concatenated short messages and provides a first level of security.

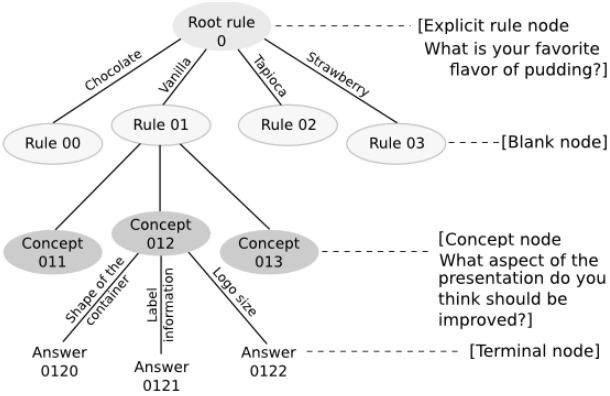


Fig. 2. moviQuest-MAS questionnaire trees

5.2 Mobile Smart Agent

Mobile smart agent is a mobile phone application whose core component is an expert system that can run on most feature phones and smartphones. The purpose of a mobile smart agent application is to convert a mobile phone into an intelligent device called a *trainedphone* (see figure 3). The concept of trainedphone is proposed as an alternative way to create intelligent mobile phone against that of smartphones: While a smartphone is manufactured with superior technology that gives it a high level of innate intelligence, a trainedphone is provided after its creation with a body of knowledge that make it more capable of having a high performance in their environment. The main task of mobile smart agents is to select the questions sequence that will be presented to the user in a real time survey, simulating a personal interview in which an expert interviewer tries to obtain the opinion from a respondent with the minimum of questions.

5.3 moviQuest-MAS Decision Making

The organizational structure used in moviQuest-MAS, where users are grouped in terms of their affinities and using a controlled registration scheme (based on private groups) provides an excellent platform to generate reliable opinions on specific issues. moviQuest-MAS also includes a population of intelligent software agents that are capable of generating complementary views to those offered by the users. Opinions given bu human agents can be obtained in an efficient way through mobile application. Thanks to these characteristics, moviQuest-MAS provides a suitable platform for group decision making. Ubiquitous Artificial Intelligence offers a method for group decision making called "moviQuest Decision Making" (MQDM) [13]. This method allows to integrate individual opinions of both human and software agents into a unique global solution. The fuzzy aggregation algorithm used in MQDM takes into account aspects such as user membership in diverse affinity groups, its record of accuracy in past decisions

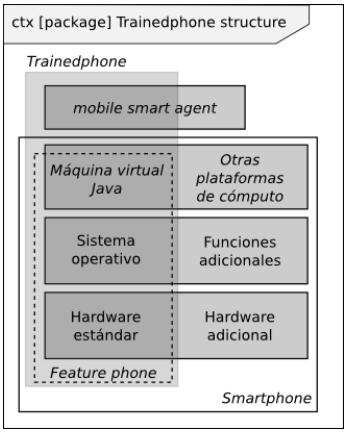


Fig. 3. *Trained phone*

and its own statement of wisdom on a given topic. The MQDM method has been designed with a perspective of sentiment analysis in order to take into account not only the direct opinion of the respondent, but also the intention embedded in its response.

6 QuoSity

moviQuest-MAS features has been tested through the social network QuoSity (http://www.moviquest.com). QuoSity is a social network system that seeks to solve some problems of interaction in educational settings. In the first instance, quosity is presented as an option for teachers and students to interact academically. Although currently there is a rich variety of software platforms for this purpose, they are often underutilized by students and teachers who prefer other freer interaction environments, such as Facebook and other social networking websites. The main constraint to the adoption of academic platforms by users is that these platforms leave out casual conversations, creating a dichotomy in the social life of the user. On the other hand the use of generic social networking platforms for academic communities causes that serious conversations (such as an academic assignment) get mixed with gossip talks. Facebook gives users the option of joining academic groups, but this service is limited to the use of institutional email addresses. Thus, the user must again separate their formal talks (held by institutional email) from his conversations with friends and family (conducted using a personal email account). In QuoSity talks are naturally separated by the groups to which the user belongs. All conversations take place in the specific context defined by these groups. Academic groups are defined by the user without further restrictions than for other groups. Also, the interaction that an organization can establish with its members through generic microblogging platforms, is restricted to the publications that both the organization and its members make. Currently,

QuoSity offers most of the functionality of moviQuest-MAS, including the use of mobile surveys and spontaneous evaluations.

7 Conclusions

moviQuest-MAS is developed around the concept of Ubiquitous Intelligence, a set of tools which allows to build artificially intelligent social networks that can offer pervasive access and high performance to users and virtual organizations.

The moviQuest-MAS software ecosystem consists of three main components: a) The control center software which gives the social network owner complete control over the system resources and improves information security risks management, b) The social network engine which provides access to the virtual community through a standard web browser and c) The mobile client through which the client communicates with the organization and the social network via SMS.

The multiagent architecture of moviQuest-MAS offers the possibility to scale the social networking capabilities and modify features in a simple way, being enough to eliminate needless agents or add new agents who carry out the new functionalities required by users.

moviQuest-MAS provides a platform for implementing social networks in which interaction is restricted to people who have any affinities with each other. This approach allows for serious relationships that are of real value both for users and for the owner of the social network.

moviQuest-MAS offers a social networking platform that can be extremely useful to help organizations to establish a ubiquitous and timely communication with the members of their community.

Acknowledgment. moviQuest-MAS is a multiagent based extension to the moviQuest platform. moviQuest was developed by the mexican firm moviQuest, S.C. holder of the copyright for the entire system and its trademark record. Multiagent modeling is being developed by moviQuest, S.C. in collaboration with the University of Sonora to be used as an umbrella model for a variety of ubiquitous business projects. The related software extension to the platform is performed by moviQuest, S.C.

References

1. Bergenti, F., Franchi, E., Poggi, A.: Agent-based Social Networks for Enterprise Collaboration. In: 2011 20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE (2011)
2. Bosch, J.: From Software Product Lines to Software Ecosystems. In: 13th International Software Product Line Conference (SPLC 2009) (August 2009)
3. Bodriagov, O., Buchegger, S.: Encryption for Peer-to-Peer Social Networks. In: 2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and 2011 IEEE International Conference on Social Computing (2011)

4. Cameron, J.J., Leung, C.K., Tanbeer, S.K.: Finding Strong Groups of Friends among Friends in Social Networks. In: 2011 Ninth IEEE Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing (2011)
5. Canali, C., Casolari, S., Lancellotti, R.: A quantitative methodology to identify relevant users in social networks. In: 2010 IEEE International Workshop on Business Applications of Social Network Analysis (BASNA) (December 2010)
6. Al Falahi, K., Atif, Y., Elnaffar, S.: Social Networks: Challenges And New Opportunities. In: 2010 IEEE/ACM International Conference on Green Computing and Communications & 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing (2010)
7. Ferber, J., Gutknecht, O., Michel, F.: From Agents to Organizations: An Organizational View of Multi-agent Systems. In: Giorgini, P., Müller, J.P., Odell, J.J. (eds.) AOSE 2003. LNCS, vol. 2935, pp. 214–230. Springer, Heidelberg (2004)
8. Lan, L., Jin, H., Lu, Y.: Personalized anonymity in social networks data publication. In: 2011 IEEE International Conference on Computer Science and Automation Engineering, CSAE (2011)
9. Maamar, Z., Badr, Y.: Social networks as a service in modern enterprises. In: 2009 International Conference on the Current Trends in Information Technology, CTIT (2009)
10. Mavridis, N.: On Artificial Agents within Human Social Networks: Examples, Open Questions, and Potentialities. In: 4th IEEE International Conference on Digital Ecosystems and Technologies, IEEE DEST 2010 (2010)
11. Neisser, U., Boodoo, G., Bouchard, T.J., Boykin, A.W., Brody, N., Ceci, S.J., Halpern, D.F., Loehlin, J.C., Perloff, R., Sternberg, R.J., Urbina, S.: Intelligence: Knowns and Unknowns. American Psychologist 51 (1996)
12. Ross, P.E.: Top 11 technologies of the decade. IEEE Spectrum 48(1) (2011)
13. Soto, R., Soto, A., Camalich, J.A.: MQDM: A Method for Fuzzy Group Decision Making in Structured Social Networks. In: The 16th IASTED International Conference on Software Engineering and Application, SEA 2012 (2012)
14. Yüksel, A.S., Yksel, M.E., Zaim, A.H.: An Approach for Protecting Privacy on Social Networks. In: 2010 Fifth International Conference on Systems and Networks Communications (2010)

Substitution Tasks Method for Discrete Optimization

Ewa Dudek-Dyduch[1] and Lidia Dutkiewicz[2]

[1] AGH University of Science and Technology,
al. Mickiewicza 30, 30-059 Cracow, Poland
edd@agh.edu.pl
[2] AGH University of Science and Technology,
al. Mickiewicza 30, 30-059 Cracow, Poland
lidia@agh.edu.pl

Abstract. The aim of the paper is to present a novel heuristic optimization method for discrete dynamic optimization problems. The method has been named substitution tasks method (ST method). According to the method, a solution is generated by means of sequence of dynamically created local optimization tasks so-called substitution tasks. The method is based on formal algebraic-logical meta model of multistage decision process (ALMM of MDP), that is given in the paper. The paper presents a formal approach for designing constructive algorithms that are based on the method. A general idea of creating substitution tasks for different optimization problems is given. Then creation of substitution tasks, based on automatic analisys of set of non-admissible states is proposed. To illustrate the presented ideas, a scheduling algorithm for a particular NP-hard problem is given and results of computer experiments are presented.

Keywords: substitution tasks method, multistage decision process, algebraic-logical meta model, scheduling problem, discrete dynamic optimization.

1 Introduction

The paper presents a new heuristic optimization method within artificial intelligence. The method, named substitution tasks method (ST method), is based on (approximately) decomposing the task of solution generation into a sequence of dynamically created substitution tasks. ST method is based on formal meta model of multistage decision process devised by Dudek-Dyduch [4].

ST method imitates the way in which a human decision-maker deals with a complex problem. When the decision-maker has to make a sequence of decisions in a long time horizon to achieve a specified goal and it is hard to predict long-term results, this decision-maker usually proceeds as described below. Instead of analyzing the whole problem, he or she tries to replace it with a few relatively simple partial problems (intermediate goals). As each partial task has a shorter time horizon, it is easier to determine suitable decisions for it. Furthermore, while

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 419–430, 2013.

solving a given partial task, additional knowledge can be collected and applied in the process. After implementing each single decision, the decision-maker reconsiders resulting situation. Based on the gained information, the partial task is updated (replaced, modified or kept as is). The procedure described above serves as a basis for the heuristic substitution tasks method presented in this paper. ST method can be, therefore, classified as artificial intelligence method. Good illustration of this idea is the way the players proceed in positional game, such as chess.

Proposed method can be used for solving many discrete, dynamic optimization problems. It is useful for control of discrete manufacturing process (e.g. tasks scheduling on multiple machines.), project management and many other combinatorial problems.

The paper extends ideas given in [8] [11] [12] [1].

The aim of the paper is 3-fold:

- to present a formal approach for designing constructive algorithms that use special local optimization tasks, so called substitution tasks, and are based on a formal algebraic-logical meta model of multistage decision process (ALMM of MDP),
- to present a general idea of creating substitution tasks for different optimization problems,
- to present an application of substitution tasks method for an NP-hard scheduling problem.

The paper is organized as follows. Section 2 presents algebraic-logical meta model that is the basis for the further consideration. In Section 3 the general idea of ST method is described. The way of creation of substitution tasks, especially that one based on automatic analisys of set of non-admissible states, is given in Section 4 . Application of ST method to a very difficult scheduling problem, namely scheduling problem with state depended resources, is described in Section 5. The section contains mathematical, algebraic-logical model of the optimization problem, algorithm based on ST method and results of computer experiments. The last section contains conclusions.

2 Algebraic-Logical Meta Model of Multistage Decision Process

Let us consider a control (planning) of any discrete deterministic process. An admissible way of the process realization can be determined with the help of simulation experiments. A single experiment establishes a sequence of decisions related to the control of the process.

Simulation course consists of determining a sequence of process states and the related time instances. The new state and its time instant depend on the previous state and the decision that had been realized (taken) then. Any discrete simulation process that is joined with decision process can be formally presented by meta model of multistage decision process devised by Dudek-Dyduch [4]. Let us recall its definition.

Definition 1. *A multistage decision process (MDP) is a process that is defined by the sextuple* $MDP = (U, S, s_0, f, S_N, S_G)$ *where* U *is a set of decisions,* $S = X \times T$ *is a set named a set of generalized states,* X *is a set of proper states,* $T \subset \mathbb{R}^+ \cup \{0\}$ *is a subset of non-negative real numbers representing the time instants,* $f: U \times S \to S$ *is a partial function called a transition function, (it does not have to be defined for all elements of the set* $U \times S$*),* $s_0 = (x_0, t_0)$*,* $S_N \subset S$*,* $S_G \subset S$ *are respectively: an initial generalized state, a set of non-admissible generalized states, and a set of goal generalized states, i.e. the states we want the process to reach in the end. The transition function is defined by means of two functions,* $f = (f_x, f_t)$ *where* $f_x: U \times X \times T \to X$ *determines the next state and* $f_t: U \times X \times T \to T$ *determines the next time instant. It is assumed that the difference* $\Delta t = f_t(u, x, t) - t$ *has a value that is both finite and positive.*

Thus, as a result of the decision u that is taken or realized at the proper state x and the moment t, the state of the process changes to $x' = f_x(u, x, t)$ that is observed at the moment $t' = f_t(u, x, t) = t + \Delta t$.

Since not all decisions defined formally make sense in certain situations, the transition function f is defined as a partial function. Thus all limitations concerning the control decisions in a given state s can be defined in a convenient way by means of so-called sets of possible decisions $U_p(s)$, and defined as: $U_p(s) = \{u \in U : (u, s) \in Domf\}$

Many different types of discrete optimization problems, especially discrete dynamic optimization problems, can be modeled by means of the above formal model. Such problems as control of discrete manufacturing processes, especially scheduling problems, project management problems and others can be formally defined in this way. Because of that, the paradigm $MDP = (U, S, s_0, f, S_N, S_G)$ is in fact a meta model of multistage decision process.

In the most general case, sets U and X may be presented as a Cartesian product $U = U^1 \times U^2 \times ... \times U^m, X = X^1 \times X^2 \times ... \times X^n$ i.e. $u = (u^1, u^2, ..., u^m)$, $x = (x^1, x^2, ..., x^n)$. Particular u^i, $i = 1, 2, ..m$ represent separate decisions that must or may be taken at the same time and relate to particular objects in the process (executors, resources, tasks etc.). There are no general limitations imposed on the sets; in particular they do not have to be numerical. The values of particular co-ordinates of a state may be names of individuals (symbols) as well as some objects (e.g. finite set, sequence etc.). The sets S_N, S_G , and U_p are formally defined with the use of logical formulae. Therefore, the complete model constitutes a specialized form of an algebraic-logical meta model.

The MDP can represent all potential possibilities of the modeled process realization. The knowledge regarding the process (all rules and constraints) is represented by U, S, s_0, f, S_N, S_G.

At the same time, a fixed MDP (i.e. when U, S, s_0, f, S_N, S_G are defined) represents a set of its trajectories that start from the initial state s_0. Let us recall that trajectory is a sequence of states $\tilde{s} = (s_0, s_1, s_2, \ldots s_i, s_{i+1}, \ldots s_k)$ where $s_{i+1} = f(u_i, s_i)$ and $s_k \in S_G \cup S_N$ or $U_p(s_k) = \emptyset$ It is assumed that no state of a trajectory, apart from the last one, may belong to the set of non-admissible generalized states S_N or may have an empty set of possible decisions $U_p(s)$.

Only a trajectory that ends in the set of goal states is admissible. The control sequence determining an admissible trajectory is an admissible decision sequence. The admissible trajectory corresponds to the admissible process realization.

The task of optimization is to find such an admissible decision sequence $\tilde{u}$ that minimizes a certain criterion Q. The optimization problem is defined by the pair (P, Q), where P represents all the constraints imposed on the process modeled by MDP.

The ALMM of MDP constitutes the basis for defining novel heuristic discrete optimization methods. Based on ALMM of MDP, learning-based method has been proposed and developed [4], [7], [10], [9], [16] as well as a method of production planning in failure modes [18]. Automatic creation of lower bounds for branch and bound method has been also worked out basing on ALMM [3]. ALMM of MDP makes it possible to define mathematic properties of discrete optimization problems. As a result, the proposed heuristic methods and algorithms can be explained and discussed formally. ST method proposed in this article will be also presented in this way.

Recently, many researchers develop ontology approach to modeling and optimizing real-life processes, especially manufacturing processes, logistic processes etc. [1] [2]. Let us notice that ALMM of MDP is also a knowledge-based meta model that combines both declarative and procedural knowledge. Thus this paper ties in with this formal line of research.

3 The Idea of the Substitution Tasks Method

Substitution tasks method is a constructive method in which whole trajectories are generated. While generating the solution (i.e. the process trajectory), in each state s of the process a decision is made on the basis of a specially constructed optimization task named substitution task $ZZ(s)$. The substitution task may be different in each state of the process. Substitution tasks are created to facilitate the decision making at a given state by substituting global optimization task with a simpler local task. After determining the best decision $u^*(s)$, the next process state s' is generated. Then, an automatic analysis of the new process state is performed and, on the basis of information gained, a new or modified substitution task is defined. Thus, in each iteration of the method, computations are performed at two levels:

1. The level of automatic analysis of the process and constructing a substitution task.
2. The level of determining possibly optimal decision for the substitution task and computing the next state.

Substitution task $ZZ(s) = (P_Z, Q_Z)$, where: P_Z - a certain substitution multistage process and Q_Z - substitution criterion. In order to highlight that the substitution process is constructed for the state s, we use the notation $P_Z(s)$. Substitution task construction presented in this paper is based upon the concept of so-called intermediate goals.

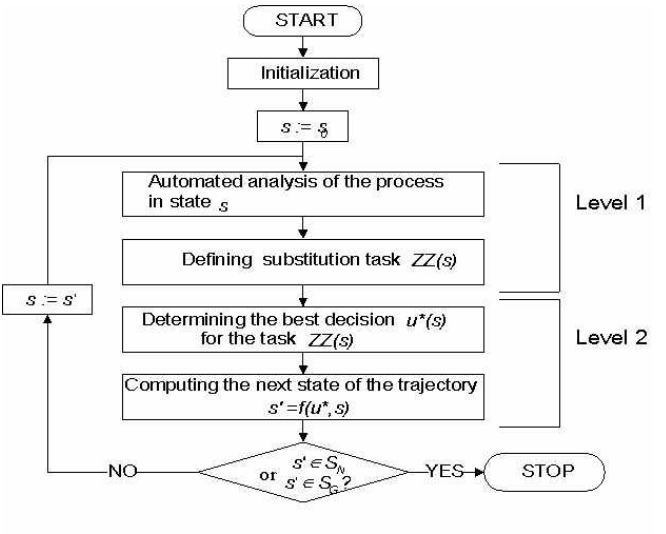


Fig. 1. Schemma of substitution tasks method

Definition 2. *Intermediate goal d is defined as achieving by the process, as soon as possible, a certain set of states* S_d.

In scheduling problems, it is most often the case that the subset of states associated with the intermediate goals consists of such states in which the distinguished task or one of the distinguished tasks is completed. The distinguished intermediate goals are used to define the set of final states of the substitution process P_Z. Thus for the substitution process a new set of final states S_{Gz} is defined, the initial state s_{0z} is the current state s of the basic process P, whilst the transition function and the sets U, S, S_N are the same as for the basic process P. As a result, the substitution process is defined as follows.

$$P_Z(s) = (U, S, s_{0z}, f, S_N, S_{Gz}) \tag{1}$$

It needs to be emphasized that the substitution task $ZZ(s)$ is for choosing only one single decision in the state s and not for determining a sequence of decisions leading the process P_Z from this state to the set of final states S_{Gz}.

4 Automatic Analysis and Creation of Substitution Tasks

The following question arises: is it possible to define how substitution tasks should be constructed for any type of problem? The answer is: yes but the problem must be presented by algebraic-logical model (ALM). Based on the analysis, the intermediate goals d_i, $i = 1, 2, \ldots$ are determined in a heuristic way. The rules for goal determining and procedures based on them strongly depend on the given optimization problem. For some problem instances, the

goals may also be defined by an expert. This article presents a method for goal determining based on the definition of the set of non-admissible states S_N. The set S_N is defined through a logical formula ϕ_N. $S_N = \{s : \phi_N(s)\}$, in particular: $\phi_N(s) = \phi_1(s) \cup \phi_2(s) \cup \ldots \phi_k(s)$, Each constraint ϕ_i is connected with a subset of non-admissible states $S(\phi_i)$ for which $\phi_i(s)$ is true. Therefore: $S_N(s) = S(\phi_1) \cup S(\phi_2) \cup \ldots S(\phi_k)$ where $S(\phi_i) = \{s : \phi_i(s)\}$ Let us notice that while generating subsequent states of the trajectory, the set of states that may be reachable by means of an admissible decision sequences undergoes changes.

Definition 3. *The subset $S_{Ach}(si)$ is achievable from the state s_i if and only if there is a sequence of decisions $(u_i, u_{i+1}, \ldots u_{i+k-1})$ such that for the generated part of the trajectory $(s_{i+1}, s_{i+2}, \ldots s_{i+k})$ the state s_{i+k} belongs to S_{Ach}. If such a sequence does not exist the subset is not achievable from s_i.*

Definition 4. *The constraint ϕ_k is active in the state s_i if the set $S(\phi_i)$ defined through that constraint is achievable from the state s_i. Otherwise, the constrain ϕ_i is inactive in a given state.*

There is a large class of problems which are characterized by the following property: if a constraint is no longer active at a certain trajectory state s, then it remains inactive in all further states of any trajectories that starts from the state s. We called this property *a permanent constraint inactivity.* A major part of the known scheduling problems has this property. For the needs of the further discussion, let us assume that the considered optimization problem possesses this property as well. The set of constraints active in a given state defines the subset $S_{NA}(s) \subseteq S_N$ and referred to as active non-admissible set. The more constraints are inactive in a given state, the smaller is the active set of non-admissible states, the higher the chances for generating an admissible trajectory. At the same time there is much more freedom in the decision-making process (higher possibility of making decisions which are advantageous in the context of the criterion).

The general idea is to generate a trajectory in such a way as to deactivate certain constraints and at the same time make the subsets of S_N inachievable in an advantageous order. Thus, intermediate goal d_i will be to achieve the subset of states S_{d_i}, for which a certain constraint ϕ_i becomes inactive. The set D will be implied by the subset of active constraints that we want to deactivate in a possibly shortest time. The priorities will define the scheduling of eliminating active constraints, that is most favorable order in which the trajectory should reach the subsets of states connected with these goals.

Recall that $ZZ = (P_Z, Q_Z)$. In order to define the substitution process P_Z, it is necessary to define a new set of goal sates S_{Gz}. This set is defined on the basis of the distinguished subset of intermediate goals $D_W(s) \subseteq D$. When choosing the subset $D_W(s)$ it is most often the case that priorities of goals are taken into consideration. Also certain additional information can be included, especially related to, for example, the current system state. The number k of intermediate goals selected for the subset $D_W(s)$ can be the same or different at each iteration

step. Afterwards, a set of final states S_{Gz} of the substitution process $P_Z(s)$ is defined as intersection of all subsets S_{d_i} defined by particular intermediate goals $d_i \in D_W(s)(i = 1, 2, \ldots k)$.

$$S_{Gz} = \bigcap_{d_i \in D_W(s)} S_{d_i} \tag{2}$$

Summing up, construction of a substitution task in a given state s is realized through the following steps: definition of the set of goals D, definition of priorities in the set of goals D, choice of the set $D_W(s)$ of goals for realization, and definition of the set of final states S_{Gz} for the substitution process P_Z.

The method of selecting decision u^* is strongly dependent upon the substitution task (see example). Due to the fact that the entire problem is formally presented as an algebraic-logical model, both analysis and substitution task creation methods can be algoritmized.

5 Scheduling Problem with State Depended Resources

To illustrate an application of the method, let us consider a specific, very difficult scheduling problem that takes place during scheduling preparatory works in mines. The set of headings in the mine must be driven in order to render the exploitation field accessible. The headings form a network, formally represented by a nonoriented multigraph $G = (W, C, R)$ where the set of branches C and the set of nodes W represent the set of headings and the set of heading crossings respectively, and relation $R \subset (W \times C \times W)$ determines connections between the headings (a partial order between the headings).

There are two kinds of machines that differ in efficiency, cost of driving and necessity of transport. Machines of the first kind (set $M1$) are more effective but the cost of driving by means of them is much higher than for the second kind (set $M2$). Additionally, the first kind of machines must be transported when driving starts from another heading crossing than the one in which the machine is, while the second type of machines need no transport. Driving a heading cannot be interrupted before its completion and can be done only by one machine at a time.

There are given deadlines for some of the headings. They result from the formerly prepared plan of field exploitation. One must determine the order of heading driving and kind of working group by means of which each heading should be driven so that the total cost of driving is minimal and each heading complete before its deadline. There are given: lengths of the headings $dl(c)$, efficiency of both kinds of machines $V_{Dr}(m)$ (driving length per time unit), cost of a length unit driven for both kinds of machines, cost of the time unit waiting for both kinds of machines, speed of machine transport $V_{Tr}(m)$ and transport cost per a length unit. Let us notice that the driven heading becomes the transport ways and may accelerate realization of other headings (tasks). Thus it is problem with state depended resources. It is strongly NP-hard.

5.1 Formal Model of Problem

The process state at any instant t is defined as a vector $x = (x^0, x^1, x^2, ..., x^{|M|})$, where $M = M1 \cup M2$. A coordinate x^0 describes a set of headings (branches) that have been driven to the moment t. The other coordinates x^m describes state of the m-th machine, where $m = 1, 2, ..., |M|$. A structure of the machine state is as follows: $x^m = (p, \omega, \lambda)$ where $p \in C \cup \{0\}$ represents the number of the heading assigned to the m-th machine to drive; $\omega \in W$ - the number of the crossing (node), where the machine is located or the number of the node, in which it finishes driving the assigned heading c; $\lambda \in [0, \infty)$ - the length of the route that remains to reach the node $\omega = w$ by the m-th machine (in particular $\lambda > dl(c)$ means that the machine is being transported to the heading, the value λ is the sum of the length of heading c and the length of the route until the transportation is finished).

A state $s = (x, t)$ belongs to the set of non-admissible states S_N if there is a heading which driving is not complete yet and its deadline is earlier than t. When we want to emphasize that a heading c has deadline, we use a notation $\hat{c}$. Let $\hat{c}_1, \hat{c}_2, \ldots \hat{c}_i, \ldots \hat{c}_k$ denote k headings that have a deadline and $\hat{C}$ - set of these headings. The definition of S_N is as follows: $S_N = \{s = (x, t) : (\exists \hat{c}_i \in \hat{C}, \hat{c}_i \notin x^0) \wedge deadline(\hat{c}_i) < t\}$, where $deadline(\hat{c}_i)$ denotes the deadline for the heading $\hat{c}_i$. Thus $S_N = S(\phi_1) \cup S(\phi_2) \cup \ldots \cup S(\phi_k)$ where $S(\phi_i) = \{s = (x, t) : \hat{c}_i \notin x^0) \wedge deadline(\hat{c}_i) < t\}$ for $i = 1, ...k$.

A state $s = (x, t)$ is a goal state if all the headings have been driven. The definition of the set of goal states S_G is as follows: $S_G = \{s = (x, t) : s \notin S_N \wedge (\forall c \in C, c \in x^0\}$. A decision determines the headings that should be started at the moment t, machines which drive, machines that should be transported, headings along which machines are to be transported and machines that should wait. Thus, the decision $u = (u^1, u^2, ..., u^{|M|})$ where the co ordinate u^m refers to the m-th machine and $u^m \in C \cup \{0\}$. $u^m = 0$ denotes continuation of the previous machine operations (continuation of driving with possible transport or further stopover). $u^m = c$ denotes the number of heading c that is assigned to be driven by machine m. As a result of this decision, the machine starts driving the heading c or is transported by the shortest way to the begining of this heading.

Obviously, not all decisions can be taken in the state (x, t). The decision $u(x, t)$ must belong to the set of possible (reasonable) decisions $U_p(x, t)$. For example, a decision $u^m = c$ is possible only when the c-th heading is neither being driven nor complete and is available, i.e. there is a way to transport machine to the one of the heading crossing adjacent to the c-th heading or machine is standing in the one of the heading crossings adjacent to the c-th heading.

Moreover, in the given state $s = (x, t)$, to each machine waiting in the heading crossing w, (it has not assigned a heading to perform), we can assign an available heading or it can be decided that it should continue to wait. However, each machine which has been previously assigned a heading and is currently driving it or it is being transported to that heading, can be only assigned to continue the current activity. Based on the current state $s = (x, t)$ and the decision u taken in this state, the subsequent state $(x', t') = f(u, x, t)$ is generated by means of the

transition function f. The complete definition of the set of the possible decision $U_p(x,t)$ as well as the transition function f will be omitted here because it is not necessary to explain the idea of the substitution tasks method. The extended model of the problem is presented in [11] [13].

5.2 Algorithm

Intermediate goals $d_i, i = 1, 2, \ldots k$, are a result of S_N subset analysis, analysis of how the decision influences the criterion and the analysis of other subsets of advantageous and disadvantageous states (different kind of such subsets for the problem above are described in [11]). Depending on the selected intermediate goals, it is possible to propose various algorithms based on substitution task method. Let us consider constraints defining the set S_N. $S_N = S(\phi_1) \cup S(\phi_2) \cup \ldots \cup S(\phi_k)$ where $S(\phi_i) = \{s = (x,t) : \hat{c}_i \notin x^0) \wedge deadline(\hat{c}_i) < t\}$ for $i = 1, \ldots k$. It is easy to see that the problem satisfies the property of permanent constraint inactivity. By definition of set S_N, intermediate goals $d_i, i = 1, 2 \ldots k$ are achieving those subsets S_{d_i}, from which the subsets of non-admissible states $S(\phi_i), i = 1, 2 \ldots k$ will not be achievable. This guarantees that the admissible trajectory (solution) will be found. Achieving the goal reduces the subset S_{NA}. Let us notice that the analysis of the formal structure of the algebraic-logical model makes it possible to determine intermediate goals through automatic inference (analysis of logical formulae).

At the same time, the earlier all headings with critical deadlines are driven the earlier it will be possible to give up faster but more expensive machines. The remaining headings can then be driven by the cheapest machine, which reduces the overall cost. As a result, it is possible to choose those decisions for which the value of the criterion is lower. Substitution criterion also implies the choice of optimal transportation routes. Intermediate goal d was then defined as achieving, in the shortest time, the subset of states $S_{d_i} \subset S$, in which the heading $\hat{c}_i$ is finished. This subset is formulated in the following way: $S_{d_i} = \{s = (x,t) : \hat{c}_i \in x^0(s)\}$ where $x^0(s)$ is the value of coordinate x^0 in the state s. Let us notice that for this set the constraint $\phi_i(s)$ becomes inactive. The optimization algorithm based on the presented approach, for the above problem, is as follows.

Step 1. Setting the initial process state: $s := s_0$

Step 2. Determining intermediate goals

In the given set of headings, headings $\hat{c}$ are searched for (that is headings for which the deadline has been defined). Each of them is associated with an intermediate goal d. The number of intermediate goals in the set D is, therefore, equal to the number of active constraints in the state s_0.

Step 3. Calculating priorities of intermediate goals

Each goal d is assigned a priority $p(d)$ which depends on the estimated time slack for accomplishing the heading $\hat{c}$ associated with this goal. The slack time is the highest time reserve calculated assuming that the heading and possible route leading to it can be driven with the help of a machine with highest performance speed. The lower the slack time for the given heading the higher the priority of the goal.

Step 4. Determining the set of intermediate goals for realization $D_W(s)$
In this set $|M|$ of intermediate goals with highest priorities is placed (where $|M|$ is the number of machines in the system). If the set D is smaller than the set M, there is $|D|$ of intermediate goals in the set $D_W(s)$.

Step 5. Determining decision for the given state
In order to make decision $u^*(s)$ in the given state s, it is necessary to define all coordinates of this decision $u(s) = (u^1, u^2, \ldots, u^{|M|})$. The task is partially decomposed by the definition of intermediate goals $d \in D_W(s)$. It is therefore possible to correlate the realization of particular goals with subsequent decision coordinates u^m, which means dedicating the realization of these goals to particular machines m. The realization of the goal means driving the particular heading $\hat{c}$. If the heading $\hat{c}$ is not available (none of its ends is connected to already driven area), it is necessary to drive a route leading to this heading. The decision for the given machine will be, therefore, either to drive the heading $\hat{c}$ or the first heading on this route. Additionally, if there is a need to transport the machine to this specific heading, the optimal transportation route is calculated. In that process, we take into consideration already driven headings and also these ones which will be completed in the nearest future.

Step 6. Calculating the next state
Having established the decision $u^*(s)$, we calculate the next process state s' using the transition function $s' = f(u^*, s)$.

Step 7. Verifying stop conditions
If the state s' belongs to goal or non-admissible states, then generating the trajectory will be terminated and the result will be stored. If this is not the case, the next step should be taken.

Step 8. Updating the set D
After determining the next state, it is necessary to update the set D of intermediate goals. What is removed from the set is every goal d, for which the relevant heading $\hat{c}$ has been completed. The number of goals in the set D is always equal to the number of active constraints.

Step 9. Initiating the next algorithm iteration
To begin the next step, it is necessary to set $s = s'$, and then to move to step 3 (updating priorities of intermediate goals).

5.3 Experiments

The algorithm proposed in this paper has been verified through a set of test networks with a number of edges (representing headings) varying from 20 to 27. The computations were made for the example with two machines. The presented algorithm generated admissible trajectories for the majority of networks, whilst the calculation time for each of these did not exceed 1 s (for Pentium IV computer). In order to evaluate the algorithm's efficiency for test networks, brute-force search has been used. The results are presented in Table 1. Based on this data it can be concluded that the proposed algorithm finds a relatively good solution very quickly.

Table 1. Comparison of the results obtained with the algorithm based on ST Method and the brute-force search algorithm (* - best result obtained until the calculations were stopped)

Network	Driving cost for ST algorithm	Driving cost for brute-search	Error	Calculation time for brute-search
A	14434,50	13880*	3,99%	30h
B	14419,20	13942,80	3,42%	20h 1min 49s
C	15202,30	14777,20	3,99%	29h 1min 55s
D	20030,90	18971,59*	5,58%	30h
E	26844,80	25880,88*	3,72%	30h

6 Conclusions

The paper presents a novel method for discrete dynamic optimization problems, named substitution tasks method (ST method). The method is a construction one, i.e. it generates whole solution (trajectory). The method is based on decomposing the task of solution generation into a sequence of dynamically created local optimization tasks (substitution tasks). It allow to simplify the decision making in each state of the optimized process. The method is based on a general formal model of multistage decision processes (ALMM of MDP), that is given in the paper. Thanks to ALMM of MDP, the way of substitute task creation is proposed and presented formally in the paper.

A large number of difficult discrete optimization problems can be efficiently solved by means of the method. Especially, the method is very useful for difficult scheduling problems with state depended resources. To illustrate the concept, some NP-hard problem, namely a scheduling problem with state depended resources, is considered and the algorithm based on ST method is presented. Results of computer experiments confirm the efficiency of the algorithm.

Recently, more and more difficult control manufacturing problems are being solved [15], [14]. Many of them have time depended constraints. Because the substitution tasks method is also suitable for the time dependent constraints the authors hope that the ST method will be implemented in manufacturing management systems.

References

1. Bocewicz, G., Wójcik, R., Banaszak, Z.A.: Reachability of Cyclic Steady States Space: Declarative Modeling Approach. In: Nguyen, N.-T., Hoang, K., Jędrzejowicz, P. (eds.) ICCCI 2012, Part II. LNCS, vol. 7654, pp. 233–243. Springer, Heidelberg (2012)
2. Bocewicz, G., Wójcik, R., Banaszak, Z.: Knowledge Base Admissibility: An Ontology Perspective. In: Grzech, A., Borzemski, L., Świątek, J., Wilimowska, Z. (eds.) Information Systems Architecture and Technology, Networks Design and Analysis, pp. 13–22. Publishing house of Wrocaw University of Technology, Wrocaw (2012)

3. Dudek-Dyduch, E.: Control of discrete event processes - branch and bound method. In: Proc. of IFAC/Ifors/Imacs Symposium Large Scale Systems: Theory and Applications, Chinese Association of Automation, vol. 2, pp. 573–578 (1992)
4. Dudek-Dyduch, E.: Formalization and Analysis of Problems of Discrete Manufacturing Processes. Scientific Bulletin of AGH University, Automatics 54 (1990) (in Polish)
5. Dudek-Dyduch, E.: Learning based algorithm in scheduling. Journal of Intelligent Manufacturing 11(2), 135–143 (2000)
6. Dudek-Dyduch, E., Dutkiewicz, L.: Substitution task method for NP-hard scheduling problems. Scientific Bulletin of Silesian University of Technology no.1726. Automatics 143, 57–66(2006) (in Polish)
7. Dudek-Dyduch, E., Dyduch, T.: Learning algorithms for scheduling using knowledge based model. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 1091–1100. Springer, Heidelberg (2006)
8. Dudek-Dyduch, E., Fuchs-Seliger, S.: Approximate algorithms for some tasks in management and economy. System, Modelling, Control 1(7) (1993)
9. Dudek-Dyduch, E., Kucharska, E.: Learning method for co-operation. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part II. LNCS, vol. 6923, pp. 290–300. Springer, Heidelberg (2011)
10. Dudek-Dyduch, E., Kucharska, E.: Optimization learning method for discrete process control. In: ICINCO 2011: Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics, vol. 1 (2011)
11. Dutkiewicz, L.: Two-Level Algorithms for Optimization of Production Processes with Resources Depending on System State. PhD thesis (2005) (in Polish)
12. Dutkiewicz, L.: Two-level algorithms for scheduling problem. Selected Problems of Computer Science. In: Rutkowska, D., et al. (eds.) Selected Problems of Computer Science, pp. 139–144. Academic Publishing House EXIT, Warsaw (2005)
13. Dutkiewicz, L., Kucharska, E.: Model of scheduling problem with state dependent resources. Semiannual: Automatics, AGH University of Science and Technology 9(1-2), 67–77 (2005)
14. Gorczyca, M., Janiak, A.: Resource level minimization in the discretre-continuous scheduling. European Journal of Operational Research 203(1) (2010)
15. Janiak, A., Janiak, W., Kovalyov, M.Y., Kozan, E., Pesch, E.: Parallel machine scheduling and common due window assignment with job independent earliness and tardiness costs. Information Sciences (in press), doi:10.1016/j.ins.2012.10.024
16. Kucharska, E., Dudek-Dyduch, E.: Extended Learning Method for Designation of Co-operation. Transactions on Computational Collective Intelligence (to appear)
17. Kucharska, E., Dutkiewicz, L.: Heuristic search of solution tree for scheduling problem with parallel machines. Semiannual: Automatics, AGH University of Science and Technology 12(3) (2008) (in Polish)
18. Sękowski, H., Dudek-Dyduch, E.: Knowledge based model for scheduling in failure modes. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 591–599. Springer, Heidelberg (2012)

Inverse Continuous Casting Problem Solved by Applying the Artificial Bee Colony Algorithm

Edyta Hetmaniok, Damian Słota, Adam Zielonka, and Mariusz Pleszczyński

Institute of Mathematics
Silesian University of Technology
Kaszubska 23, 44-100 Gliwice, Poland
{edyta.hetmaniok,damian.slota,
adam.zielonka,mariusz.pleszczynski}@polsl.pl

Abstract. The paper presents an application of the Artificial Bee Colony algorithm in solving the inverse continuous casting problem consisted in reconstruction of selected parameters characterizing the cooling conditions in crystallizer and in secondary cooling zone. In presented approach we propose to use the bee algorithm for minimization of appropriate functional representing the crucial part of the method.

Keywords: Artificial Intelligence, Swarm Intelligence, ABC Algorithm, Inverse Continuous Casting Problem.

1 Introduction

Artificial intelligence became in recent times an important branch of computer science. Under this expression we understand the systems imitating the intelligent human behavior, ability of learning, constant improvement of activity in order to maximize the chances for success [1]. One of possibilities used in that kind of mechanisms is the partition of general problem into a set of small subproblems solving of which leads to a general success. This rule constitutes the basic idea of Artificial Bee Colony (ABC) algorithm where the collective actions of artificial bees, determining the individual solutions of considered problem, compose the general solution [2, 3].

Artificial Bee Colony algorithm has been presented in the recent decade and it belongs to the group of swarm intelligence algorithms imitating the mechanisms of natural collective behaviors from the real world. Concept of using such inspiration in artificial conditions was introduced by Gerardo Beni and Jing Wang in 1989 in investigation of the cellular robotic systems [4, 5]. Idea of swarm intelligence basis on the conception of integrating some sets of relatively simple rules for simulating more complicated behaviors of the groups of animals, bacteria, plants or even human beings [6–9]. Particular individual, taking part in the procedure, is not aware of the finite goal, but by carrying out its own small task and by informing, in some way, the other members of the group about the achieved result it contributes in the success of entire swarm.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 431–440, 2013.

Artificial Bee Colony algorithm imitates the technique of searching for the nectar around the hive and communication between bees with the aid of a special dance, called the waggle dance. During this dance a part of bees, named as bees-scouts, inform the other part of bees, named as bees-viewers, about the possible locations of good sources of food. After that the bees-viewers fly to explore more precisely these pointed places and the bees-scouts, after spending some time in the hive, fly out again to search some new sources. In this way the considered space is double explored – generally at the beginning and then more carefully around the best selected locations.

Optimization technique based on this natural behavior of bees was already applied for solving a different kind of optimization problems, like for example the transportation problem, reaction-diffusion problem, generalized assignment problem and others [10–12]. Authors of the current paper used also the ABC algorithm for solving the selected inverse heat conduction problems [13–15]. Whereas in this paper we intend to propose a method based on the ABC algorithm for solving the inverse continuous casting problem. Continuous casting is a technical process during which the the molten metal is poured in the controlled way into the crystallizer where it solidifies by taking the appropriate form and next is consecutively moved out from there. Mathematically the process can be described by means of the Stefan problem [16–23]. However in our paper we will try to solve the inverse continuous casting problem consisted in reconstruction of the cooling conditions in crystallizer and in secondary cooling zone. This missing information will be compensated by the measurements of temperature taken from the sensors located in selected points of considered region. Difficulty in solving problems of that kind results from the fact that usually their analytic solutions do not exist or they exist but are neither unique nor stable [24]. Therefore the inverse problems are often called as the ill posed problems and each proposition of successful method enabling to determine the approximate solution of such problems is desired and notable.

2 Conception of the ABC Algorithm

Let us consider the function $F(\mathbf{x})$ defined in domain D without any specific assumptions. Points of the domain – vectors $\mathbf{x}$ - play the role of the sources of nectar and value of the function in given point – number $F(\mathbf{x})$ – designates the quality of source $\mathbf{x}$. Since we want ot minimize the objective function, the smaller is the value $F(\mathbf{x})$, the better is the source $\mathbf{x}$.

In the first part of the algorithm the bees-scouts explore the domain and type some number of points – candidates for the sources of nectar. Every scout makes some control movements around the selected point in order to check whether any better source can be found in the neighborhood. After that the scouts return to the hive and wait there for the next cycle. In the second part of the algorithm the bees-viewers select the sources, with the given probabilities, from among the sources discovered by the scouts in the first part. Probability of the choice of a source is the greater, the better is the quality of this source. After that the

viewers explore the selected points by making again some control movements around. The operation ends by choosing the best point – the best source of nectar – in the current cycle. Details of ABC algorithm are presented below.

Initialization of the algorithm

1. Initial data:
   SN – number of the explored sources of nectar (number of the bees-scouts, number of the bees-viewers);
   D – dimension of the source $\mathbf{x}_i$, $i = 1, \ldots, SN$;
   lim – number of "corrections" of the source position $\mathbf{x}_i$;
   MCN – maximal number of cycles.
2. Initial population – random selection of the initial sources localization represented by D-dimensional vectors $\mathbf{x}_i$, $i = 1, \ldots, SN$.
3. Calculation of values $F(\mathbf{x}_i)$, $i = 1, \ldots, SN$, for the initial population.

Main algorithm

1. Modification of sources localizations by the bees-scouts.
   a) Every bee-scout modifies position $\mathbf{x}_i$ according to the formula:
   $$v_i^j = x_i^j + \phi_{ij}(x_i^j - x_k^j), \qquad j \in \{1, \ldots, D\},$$
   where: $\left.\begin{array}{l} k \in \{1, \ldots, SN\}, k \neq i, \\ \phi_{ij} \in [-1, 1]. \end{array}\right\}$ – randomly selected numbers.
   b) If $F(\mathbf{v}_i) \leq F(\mathbf{x}_i)$, then position $\mathbf{v}_i$ replaces $\mathbf{x}_i$. Otherwise, position $\mathbf{x}_i$ stays unchanged.
   Steps a) and b) are repeated lim times. We take: $lim = SN \cdot D$.
2. Calculation of probabilities P_i for the positions $\mathbf{x}_i$ selected in step 1. We use the formula:
   $$P_i = \frac{fit_i}{\sum\limits_{j=1}^{SN} fit_j}, \quad i = 1, \ldots, SN,$$
   where: $fit_i = \begin{cases} \frac{1}{1+F(\mathbf{x}_i)} & \text{if } F(\mathbf{x}_i) \geq 0, \\ 1 + |F(\mathbf{x}_i)| & \text{if } F(\mathbf{x}_i) < 0. \end{cases}$
3. Every bee-viewer chooses one of the sources $\mathbf{x}_i$, $i = 1, \ldots, SN$, with probability P_i. Of course one source can be chosen by a group of bees.
4. Every bee-viewer explores the chosen source and modifies its position according to the procedure described in step 1.
5. Selection of the $\mathbf{x}_{best}$ for the current cycle – the best source among the sources determined by the bees-viewers. If the current $\mathbf{x}_{best}$ is better that the one from the previous cycle, we accept it as the $\mathbf{x}_{best}$ for the entire algorithm.

6. If in step 1 the bee-scout did not improve position $\mathbf{x}_i$ ($\mathbf{x}_i$ did not change), it leaves the source $\mathbf{x}_i$ and moves to the new one by using the formula:

$$x_i^j = x_{min}^j + \phi_{ij}(x_{max}^j - x_{min}^j), \qquad j = 1, \ldots, D,$$

where: $\phi_{ij} \in [0, 1]$.

Steps 1–6 are repeated MCN times.

3 Formulation of Problem

Let us have the vertical device, schematically presented in Figure 1*a*, working in the undisturbed cycle in which the continuous casting of pure metals takes place. We consider the apparently steady field of temperature generated in this process. According to the fact that the amount of heat conducted in direction of the ingot move in comparison with the amount of heat conducted in the direction perpendicular to the ingot axis is slight [16] we can take two assumptions. Firstly we assume that the cooling conditions, changing with reference to direction of the ingot forming, are identical in the entire perimeter of the ingot and the dimensions of the ingot cross section satisfy condition $a \ll d$, where a denotes the ingot thickness and $2d$ describes its width. Secondly we assume that the heat flows only in the direction perpendicular to the ingot axis.

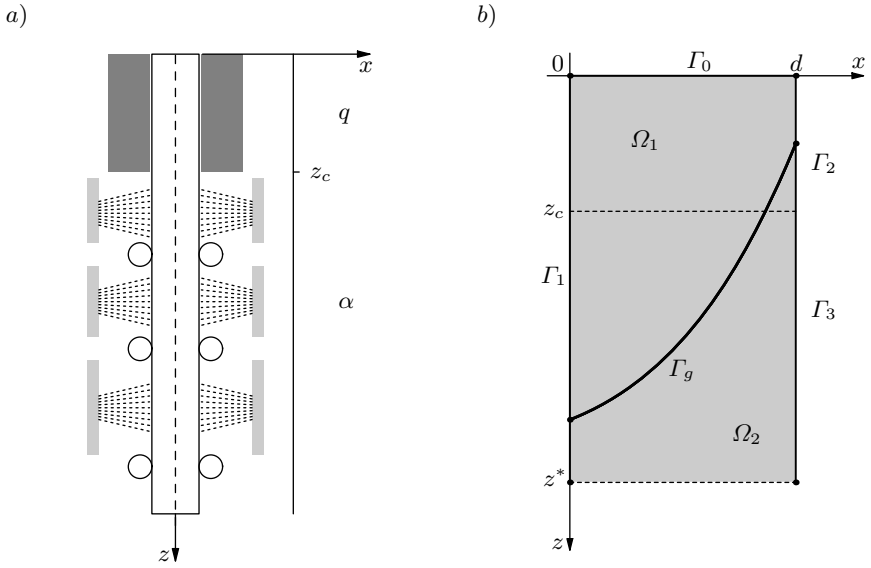


Fig. 1. Scheme of the problem (*a*) and domain of the two-dimensional problem (*b*)

Because of the heat symmetry and taken assumptions the region Ω of the ingot can be considered as the two-dimensional region consisted of two subregions: Ω_1 taken by the liquid phase and Ω_2 taken by the solid phase, separated by the

freezing front Γ_g (described by means of function $x = \xi(t)$). Taking the space orientation as in Figure 1*b*, the heat transfer process, including the apparently steady field of temperature and location of the freezing front, can be described in considered subregions with the aid of two-phase Stefan problem [16].

Boundary of region $\Omega = [0, d] \times [0, z^*] \subset \mathbb{R}^2$ is divided into four subsets (Figure 1*b*), where the boundary conditions are defined

$$\Gamma_0 = \{(x, 0);\ x \in [0, d]\}, \tag{1}$$
$$\Gamma_1 = \{(0, z);\ z \in [0, z^*]\}, \tag{2}$$
$$\Gamma_2 = \{(d, z);\ z \in [0, z_1]\}, \tag{3}$$
$$\Gamma_3 = \{(d, z);\ z \in (z_1, z^*]\}. \tag{4}$$

Functions of temperature T_k within the regions Ω_k (for $k = 1, 2$) satisfy the heat conduction equation

$$c_k\, \varrho_k\, w\, \frac{\partial T_k}{\partial z}(x, z) = \frac{\partial}{\partial x}\left(\lambda_k\, \frac{\partial T_k}{\partial x}(x, z)\right), \tag{5}$$

where c_k, ϱ_k and λ_k denote, respectively, the specific heat, mass density and thermal conductivity in liquid phase ($k = 1$) and solid phase ($k = 2$), w is the velocity of continuous casting, and, lastly, x and z denote the spatial variables.

Moreover we assume the following boundary conditions on the respective parts of the boundaries – on boundary Γ_0 the boundary condition of the first kind with the given pouring temperature ($T_z > T^*$):

$$T_1(x, 0) = T_z, \tag{6}$$

on boundary Γ_1 the homogeneous boundary condition of the second kind

$$\frac{\partial T_k}{\partial x}(0, z) = 0, \tag{7}$$

on boundary Γ_2 (crystallizer) the boundary condition of the second kind

$$-\lambda_k \frac{\partial T_k}{\partial x}(d, z) = q(z), \tag{8}$$

on boundary Γ_3 (secondary cooling zone) the boundary condition of the third kind

$$-\lambda_k \frac{\partial T_k}{\partial x}(d, z) = \alpha(z)\left(T_k(d, z) - T_\infty\right) \tag{9}$$

and, finally, on the interface Γ_g the continuity condition and the Stefan condition

$$T_1\big(\xi(z), z\big) = T_2\big(\xi(z), z\big) = T^*, \tag{10}$$
$$L\, \varrho_2\, w\, \frac{d\xi(z)}{dz} = -\lambda_1\, \frac{\partial T_1(x, z)}{\partial x}\bigg|_{x=\xi(z)} + \lambda_2\, \frac{\partial T_2(x, z)}{\partial x}\bigg|_{x=\xi(z)}, \tag{11}$$

where α describes the heat transfer coefficient, q denotes the heat flux density, T_z is the pouring temperature, T_∞ is the ambient temperature, T^* is the solidification temperature, L describes the latent heat of fusion.

Discussed problem consists in determination of the cooling conditions for the ingot in such way that the temperature in selected points of the solid phase would take the given values ($(x_i, z_j) \in \Omega_2$):

$$T_2(x_i, z_j) = U_{ij}, \qquad i = 1, 2, \ldots, N_1, \quad j = 1, 2, \ldots, N_2, \tag{12}$$

where N_1 denotes the number of sensors and N_2 describes the number of measurements taken from each sensor. The sought cooling conditions are described by the heat flux in crystallizer and the heat transfer coefficient in the secondary cooling zone, it means the following function f should be identified

$$f(z) = \begin{cases} q(z) & \text{for } z \le z_c, \\ \alpha(z) & \text{for } z > z_c. \end{cases} \tag{13}$$

For the fixed form of function f problem (5)–(11) turns into the direct Stefan problem, solving of which enables to find the courses of temperature $T_{ij} = T_2(x_i, z_j)$ corresponding to function f. By using the calculated temperatures T_{ij} and the given temperatures U_{ij} the following functional can be constructed

$$J(f) = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \left(T_{ij} - U_{ij}\right)^2, \tag{14}$$

representing the error of approximate solution. By minimizing the above functional we intend to find such form of function f that the reconstructed temperatures will be as close as possible to the measurement values. For minimizing this functional we will apply the Artificial Bee Colony algorithm and for solving the direct Stefan problem, associated with considered inverse problem, we will use the alternating phase truncation method [16, 22, 23]. Let us notice that each running of the procedure requires to solve the direct Stefan problem for many times.

4 Numerical Verification

As the exemplary process let us consider the continuous casting of aluminium described by the following parameters: $d = 0.1$ [m], $\lambda_1 = 104$ [W/(m K)], $\lambda_2 = 204$ [W/(m K)], $c_1 = 1290$ [J/(kg K)], $c_2 = 1000$ [J/(kg K)], $\varrho_1 = 2380$ [kg/m^3], $\varrho_2 = 2679$ [kg/m^3], $L = 390000$ [J/kg], velocity of casting $w = 0.002$ [m/s], solidification temperature $T^* = 930$ [K], ambient temperature $T_\infty = 298$ [K] and pouring temperature $T_z = 1013$ [K].

We need to identify the heat flux in crystallizer and the heat transfer coefficient in the secondary cooling zone, exact values of which are known:

$$q(z) = 400000 \ \ [\mathrm{W/m^2}],$$
$$\alpha(z) = 4000 \ \ [\mathrm{W/(m^2\,K)}].$$

We use two thermocouples ($N_1 = 2$) located 0.01 and 0.02 m away from boundary of considered region. From each thermocouple we take 100 measurements of temperature ($N_1 = 100$). Distance along the Oz axis between the successive measurements is equal to 0.002 m. In calculations we used the exact values of temperature and values burdened by the 1, 2 and 5% random error of normal distribution.

ABC algorithm was executed for number of bees $SN = 8$ and number of cycles $MCN = 15$. Because of the big difference between expected values of reconstructed parameters the initial locations of sources were randomly selected from different intervals (for reconstructing q from interval $[250000, 500000]$ and for reconstructing α from interval $[1000, 5000]$). Specific quality of all the swarm intelligence algorithms, including ABC, is their heuristic nature which means that each execution of the procedure can give slightly different results. Therefore to assure the best results we evaluated the calculations for 20 times and the best of received results were taken as the reconstructed elements.

Figure 2 present the relative errors of the heat flux q and the heat transfer coefficient α identification obtained for input data burdened by 2% and 5% error, respectively, displayed in dependence on the number of cycles of the procedure. In both cases the reconstruction errors are smaller than the input data errors. Let us also notice that satisfying results are achieved very quickly and the further executions of the procedure do not improve significantly the results. Distributions of error received for the unburdened input data are note presented here because of the limited size of the paper, however they converge very quickly to zero which confirms the stability of used procedure.

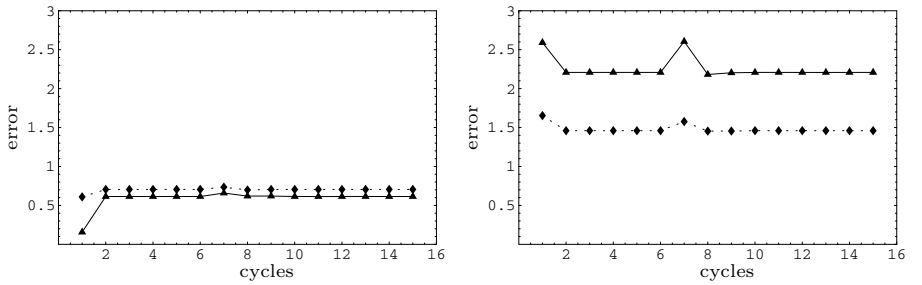


Fig. 2. Relative errors of parameter f reconstruction for the successive iterations (▲ – for q, ♦ – for α) obtained for 2% (left figure) and 5% (right figure) noise of input data

In Figures 3 and 4 the reconstructed distributions of temperature in measurement points located, respectively, 0.01 m and 0.02 m away from boundary of the region in comparison with the known exact distributions are showed. The results are obtained for input data burdened by 5% error and one can observe that for both locations the reconstructed and known courses of temperature

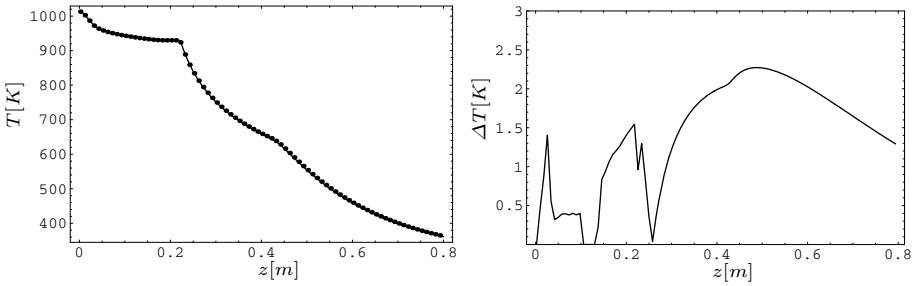


Fig. 3. Exact (solid line) and reconstructed (dots) distributions of temperature (left figure) in control point located 0.01 m away from boundary obtained for 5% noise of input data and absolute error of this reconstruction (right figure)

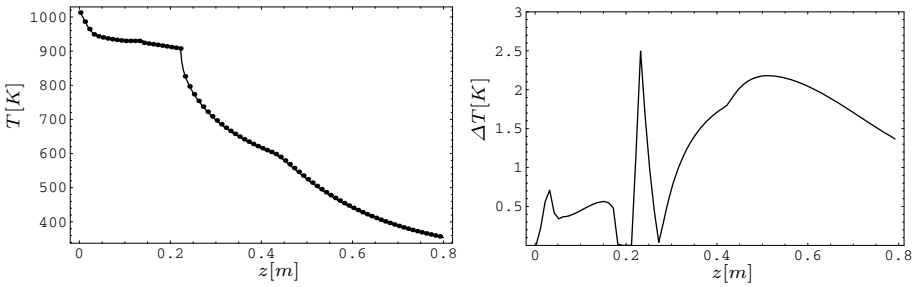


Fig. 4. Exact (solid line) and reconstructed (dots) distributions of temperature (left figure) in control point located 0.02 m away from boundary obtained for 5% noise of input data and absolute error of this reconstruction (right figure)

almost cover. Figures present also the absolute errors of these reconstructions. In both cases the absolute errors do not exceed 2.5 K, therefore the reconstruction of temperature is almost perfect.

Above figures show the results obtained in one execution of the procedure selected from among 20 executions. To confirm stability of the method let us present the statistical elaboration of results obtained in 20 executions for various noises of input data compiled in Table 1. Relative errors of the heat flux and the heat transfer coefficient reconstruction for unburdened input data are smaller than 0.1% and certainly they increase with the increasing value of input data perturbation, however they are always much smaller than the input data error. The maximal relative error of temperature reconstruction in each considered case is slight. Differences between obtained results are small – the standard deviation of the sought parameters reconstruction in the worst case is at the level of 2% of the mean value of this reconstruction.

Table 1. Reconstructed values of parameter f for boundary condition, relative errors (δ_f), standard deviations (σ) and standard deviations (σ^p) expressed as a percent of mean values of parameter f reconstruction together with the maximal relative errors (δ_T^{max}) of temperature reconstruction obtained for various noises of input data

noise	f	δ_f [%]	σ	σ^p [%]	δ_T^{max}[%]
0%	400380.26	0.09507	380.262	0.09507	$2.75 \cdot 10^{-4}$
	4001.56	0.03902	1.561	0.03930	
1%	398669.20	0.33270	1330.771	0.33270	$5.04 \cdot 10^{-3}$
	3997.77	0.05577	2.231	0.05571	
2%	397543.05	0.61424	2456.942	0.61424	$5.24 \cdot 10^{-3}$
	4028.20	0.70512	28.205	0.70512	
5%	408832.51	2.20814	8832.553	2.20814	$6.13 \cdot 10^{-3}$
	3941.62	1.45966	58.386	1.45965	

5 Conclusions

In this paper we have proposed a procedure for identifying the cooling conditions in continuous casting process such that the reconstructed values of temperature are as close as possible to the measurement values taken in selected points of considered region. Presented procedure was based on the Artificial Bee Colony algorithm thanks to which the minimization of appropriate functional, representing differences between the approximate and exact temperatures, was possible. Executed experimental verification of elaborated method showed that the approximate solution rapidly converges to exact solution and the reconstruction errors are in all considered cases much smaller than the error of input data. Moreover, the satisfying results were obtained after just few cycles and further executions of the procedure did not improve significantly the results. Therefore, to improve the results probably some other techniques should be used, which is planned for the future, however in this moment we find obtained reconstructions as satisfying.

Acknowledgements. This project has been financed from the funds of the National Science Centre granted on the basis of decision DEC-2011/03/B/ST8/06004.

References

1. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Englewood Cliffs (1995)
2. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Appl. Soft Computing 8, 687–697 (2007)
3. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. Appl. Math. Comput. 214, 108–132 (2009)
4. Eberhart, R.C., Shi, Y., Kennedy, J.: Swarm Intelligence. Morgan Kaufmann, San Francisco (2001)
5. Beni, G., Wang, J.: Swarm intelligence in cellular robotic systems. In: Proceed. NATO Advanced Workshop on Robots and Biological Syst., Tuscany (1989)

6. Chu, S.-C., Tsai, P.-w., Pan, J.-S.: Cat swarm optimization. In: Yang, Q., Webb, G. (eds.) PRICAI 2006. LNCS (LNAI), vol. 4099, pp. 854–858. Springer, Heidelberg (2006)
7. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control System Magazine 22, 52–67 (2002)
8. Mehrabian, R., Lucas, C.: A novel numerical optimization algorithm inspired from weed colonization. Ecological Informatics 1(4), 355–366 (2006)
9. Geem, Z.W.: Improved Harmony Search from ensemble of music players. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4251, pp. 86–93. Springer, Heidelberg (2006)
10. Lučič, P., Teodorovič, D.: Computing with Bees: attacking complex transportation engineering problems. Int. J. Artificial Intelligence Tools 12, 375–394 (2003)
11. Tereshko, V.: Reaction-diffusion model of a honeybee colony's foraging behaviour. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 807–816. Springer, Heidelberg (2000)
12. Özbakir, L., Baykasoğlu, A., Tapkan, P.: Bees algorithm for generalized assignment problem. Appl. Math. Comput. 215, 3782–3795 (2010)
13. Hetmaniok, E., Słota, D., Zielonka, A.: Solution of the inverse heat conduction problem by using the ABC algorithm. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 659–668. Springer, Heidelberg (2010)
14. Zielonka, A., Hetmaniok, E., Słota, D.: Using of the Artificial Bee Colony Algorithm for determining the heat transfer coefficient. In: Czachórski, T., Kozielski, S., Stańczyk, U. (eds.) Man-Machine Interactions 2. Advances in Intelligent Systems and Computing, vol. 103, pp. 369–376. Springer, Heidelberg (2011)
15. Hetmaniok, E., Słota, D., Zielonka, A., Wituła, R.: Comparison of ABC and ACO algorithms applied for solving the inverse heat conduction problem. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) EC 2012 and SIDE 2012. LNCS, vol. 7269, pp. 249–257. Springer, Heidelberg (2012)
16. Mochnacki, B., Suchy, J.: Numerical Methods in Computations of Foundry Processes. PFTA, Cracow (1995)
17. Grever, W., Binder, A., Engl, H., Mörhwald, K.: Optimal cooling strategies in continuous casting of steel with variable casting speed. Inverse Probl. Eng. 2, 289–300 (1997)
18. Constalas, D., Kačur, J., Van Keer, R.: On optimal cooling strategy for variable-speed continuous casting. Int. J. Numer. Methods Engrg. 53, 539–565 (2002)
19. Nowak, I., Nowak, A.J., Wrobel, L.C.: Identification of phase change fronts by Bezier splines and BEM. Int. J. Therm. Sci. 41, 492–499 (2002)
20. Santos, C.A., Spim, J.A., Garcia, A.: Mathematical modeling and optimization strategies (genetic algorithm and knowledge base) applied to the continuous casting of steel. Eng. Appl. Artif. Intel. 16, 511–527 (2003)
21. Santos, C.A., Garcia, A., Frick, C.R., Spim, J.A.: Evaluation of heat transfer coefficient along the secondary cooling zoones in the continuous casting of steel billets. Inverse Probl. Sci. Eng. 14, 687–700 (2006)
22. Słota, D.: Identification of the cooling condition in 2-D and 3-D continuous casting processes. Numer. Heat Transfer B 55, 155–176 (2009)
23. Słota, D.: Restoring boundary conditions in the solidification of pure metals. Comput. & Structures 89, 48–54 (2011)
24. Beck, J.V., Blackwell, B., St.Clair, C.R.: Inverse Heat Conduction: Ill Posed Problems. Wiley Intersc., New York (1985)

Stock Trading with Random Forests, Trend Detection Tests and Force Index Volume Indicators

Piotr Ładyżyński[1], Kamil Żbikowski[2], and Przemysław Grzegorzewski[1,3]

[1] Faculty of Mathematics and Computer Science
Warsaw University of Technology
Koszykowa 75, 00-662 Warsaw, Poland
[2] Faculty of Electronics and Information Technology
Warsaw University of Technology
Plac Politechniki 1, 00-661 Warsaw, Poland
[3] Systems Research Institute, Polish Academy of Sciences
Newelska 6, 01-447 Warsaw, Poland
{p.ladyzynski,pgrzeg}@mini.pw.edu.pl, kamil.zbikowski@ii.pw.edu.pl

Abstract. The goal of this paper is to investigate if the strong machine learning technique is able to retrieve information from past prices and predict price movements and future trends. The architecture of the system with the on-line adaptation ability to non-stationary two dimensional mixed Black-Scholes Markov time series model is presented. The methodology of investment strategies performance verification is also proposed.

Keywords: Stock trading, random forest, trend detection test, financial time serie, quant fund, investment strategies backtesting.

1 Introduction

Since the early beginning of financial markets investors, speculates and gamblers try to predict stocks price movements and dream on the universal trading strategy generating outstanding profits. Many research have been done in the field and the annual sales of trading courses and tutorials are estimated in billion of dollars. However, a little has been done in the area of scientific research related to that field and the most of the described strategies are not verified at real financial markets.

During last years we have conducted some research of algorithmic trading techniques and investigated a lot of strategies proposed in the literature, tutorials for broker clients and trading courses. To our surprise none of these strategies was able to generate profitable signals while tested by the walk forward method, described in this paper.

A lot of publications reveal serious lacks in verification methodology and some of them seem to be only marketing tricks. There are some papers investigating usability of the machine learning techniques for automated stock trading, but

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 441–452, 2013.

none of them incorporates the walk forward testing on large amount of observations (see [3], [5], [8], [9]). It is obvious that even if a system is tested on out-of sample data (but without walk forward testing), it is easy to choose such a period that the system gives outstanding results. Therefore, although the papers mentioned above consider interesting data mining systems, it is highly probable that walk forward testing would reveal their weakness in long term trading at the real market.

Many research show that time series created by prices change their structure in time and reveal strong non-stationary behavior. Therefore, many classical time series models (like ARCH, AR etc.) are useless for the analysis of real price movements (of course, it does not contradict that some time series model are brilliant for other features of price trajectory estimation, like GARCH model for market volatility). The major purpose of this paper is to make a step towards to answer the question if one of the strongest classifiers, known in the literature, is able to generate profitable trading signals at real market.

In this paper we propose an architecture of the system which monitors the behavior of the upcoming data and prove its ability of on-line adaptation to non-stationary time series. The performance of the system is examined using large scale historical data from S&P 500 index by the moving window walk forward methodology, which was never done in the past. To be more precise, in our approach the trading system is examined by the walk forward testing on 50 randomly chosen shares from S& P 500 index during the 7 years period (2004-2011). We use one hour aggregates (all previous authors used one day aggregates) so the systems is verified on $50 \times 15000 = 750000$ stock quotations. The authors of the most trading publication verify their systems on one to five years period using day aggregation, so their data range is about $5 \times 250 = 1250$. Unfortunately, a verification of the system on such short period is not reliable. Moreover, some authors pre-tune their systems to the test data set (see, e.g. [8]). The drawbacks of such verification are obvious and described in section 3.

Hence the paper deals with extremely interesting and extensively examined hypothesis whether the machine learning techniques is able to capture the market structure for constructing a profitable trading strategy. As it is known from the history of finance that famous traders become famous due to trading books they have written after loosing large amount of money. One may ask if any human being is able to predict permanently price movements or maybe market - by the definition - is a chaotic system only with the poor property of hetreoskedasticity.

2 System Architecture

Many investors believe that certain combination of technical analysis indicators contain an important clue for future price movements prediction. Such indicators are designed to aggregate some special features of financial time series. We would try to check if the strong machine learning technique is able to retrieve trading patterns from preprocessed by indicators past prices and volume.

2.1 Trend Detection Test

In the most trading systems early trend detection is crucial. Let $(X_i)_{i=1}^n$ be a sequence of prices and let

$$m_i = \frac{X_i - X_{i-1}}{X_{i-1}} \tag{1}$$

denote the rate of returns ($i = 1, \ldots, n$). We try to detect trend applying the exponential averaging to the time series of rate of returns, i.e.

$$m_{i+1}^{(exp)} = \lambda m_{i+1} - (1 - \lambda) m_i^{(exp)}, \tag{2}$$

where $\lambda = \frac{2}{n+1}$ is a smoothing parameter. The greater values of λ the lesser weights of the observations from the past and greater weights of the recent observations.

The test statistic for trend detection is than defined as follows (see [7])

$$T^{(exp)} = m_n^{(exp)}. \tag{3}$$

The null hypothesis distribution of (3) is calculated numerically using the Monte Carlo method.

2.2 Indicators

The aggregation operators used in technical analysis are called *indicators*. We may distinguish three main groups of indicators: trend following indicators, oscillators and volume indicators.

Trend Following Indicators. Trend following indicators capture trend after it occurs on the market. Investors believe that the current trend continuation is more probable than the reversal trend and make orders according to the direction of the trend. The most important trend following indicators are:

- $p_{k,n}^{val}$, i.e. p-value of trend detection test statistics (3) calculated for day k and rate of returns window of length $n + 1$;

- the simple *moving average* of rate of returns in window of length d calculated for day k, i.e.

$$MA_k^d = \frac{1}{d} \sum_{i=0}^{d} m_{k-i}. \tag{4}$$

Oscillators. Oscillators try to guess the future trend direction before it occurs on the market. One of the most popular and useful oscillator is the *relative strength index* defined by

$$RSI_k^d = 100 - \frac{100}{1 + RS_k^d}, \tag{5}$$

where

$$RS_k^d = \frac{\sum_{i=0}^{d-1}(X_{k-i} - X_{k-i-1})\mathbb{I}_{(x\geq 0)}(X_{k-i} - X_{k-i-1})}{-\sum_{i=1}^{d-1}(X_{k-i} - X_{k-i-1})\mathbb{I}_{(x<0)}(X_{k-i} - X_{k-i-1})}. \tag{6}$$

By dividing positive and negative price differences the relative strength index tries to capture a market strength.

Another interesting operator is the *Williams oscillator*, which tries to capture the range of the past prices in comparison with the current one, i.e.

$$WL_k^d = 100\frac{\max\{X_k, X_{k-1}, \ldots, X_{k-d-1}\} - X_k}{\max\{X_k, X_{k-1}, \ldots, X_{k-d-1}\} - \min\{X_k, X_{k-1}, \ldots, X_{k-d-1}\}}. \tag{7}$$

Volume Indicators. Volume indicators analyze additional information about the market condition – a transaction volume. By combining price differences with volume levels we are able to construct operators indicating current demand and supply on the market. Let define

The *force index*, introduced by Alexander Elder (see [1]), is given by

$$FI_k = \lambda FI_k^1 + (1-\lambda)FI_{k-1} \tag{8}$$

where

$$FI_k^1 = vol_k(X_k - X_{k-1}) \tag{9}$$

and where traditionally $\lambda = \frac{2}{14}$. Thus, (8) operator indicates market's overbought and oversold moment.

2.3 Forest Learning

The essence of the proposed system is the application of random forests for constructing regression function predicting maximum and minimum prices in the next 28 hours (i.e. 4 trading days).

Let us define the *maximum relative rate of return* H_{ri}^n and the *minimum relative rate of return* L_{ri}^n in next n periods, respectively,

$$H_{ri}^n = \frac{X_i - H_i^n}{X_i} \tag{10}$$

$$L_{ri}^n = \frac{X_i - L_i^n}{X_i}, \tag{11}$$

where $H_i^n = \max\{X_{i+1}, X_{i+2}, \ldots, X_{i+n}\}$ and $L_i^n = \min\{X_{i+1}, X_{i+2}, \ldots, X_{i+n}\}$.

Our goal is to find functions f_l, f_h predicting future price levels, i.e.

$$\begin{aligned}\hat{H}_{ri}^n = f_h(\ & x_i^{(1)}, x_{i-1}^{(1)}, \ldots, x_{i-k}^{(1)}, \\ & x_i^{(2)}, x_{i-1}^{(2)}, \ldots, x_{i-k}^{(2)}, \\ & \ldots \\ & x_i^{(n)}, x_{i-1}^{(n)}, \ldots, x_{i-k}^{(n)})\end{aligned} \tag{12}$$

$$\hat{L}^n_{ri} = f_l(\ x^{(1)}_i, x^{(1)}_{i-1}, \ldots, x^{(1)}_{i-k}, \\ x^{(2)}_i, x^{(2)}_{i-1}, \ldots, x^{(2)}_{i-k}, \\ \ldots \\ x^{(n)}_i, x^{(n)}_{i-1}, \ldots, x^{(n)}_{i-k}), \tag{13}$$

where i stands for the current hour and $x^{(s)}_l$ is a value of indicator s at hour l $(l < i)$.

The structure presented above provides the classifier with the market snapshots obtained for time moments $t = i, i-1, \ldots, i-k$.

To solve a regression problem we use the following random forest algorithm (see [4]):

1. From $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbb{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b on bootstrapped sample by recursively repeating the following steps for each node of the tree, until minimum size of the node size n_{min} is reached:
      i. Select m variables at random from p variables from feature sample.
      ii. Pick the best variable/split-point among m.
      iii. Split the node into two child nodes.
2. Output the forest: $\{T_b\}_{b=1}^B$

To predict a new point x we get

$$\hat{f}^B_{rf}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{14}$$

Now, using the p-value of the trend detection test for three different time horizons, and technical analysis indicators, we construct feature matrices for high and low prices regression models:

$$\begin{array}{cccccc} y & \bar{x} & & & & \\ H^n_{rk} & r_k & r_{k-1} & \ldots & r_{k-n} \\ & p^{val}_{k,n_1} & p^{val}_{k-1,n_1} & \ldots & p^{val}_{k-n,n_1} \\ & p^{val}_{k,n_2} & p^{val}_{k-1,n_2} & \ldots & p^{val}_{k-n,n_2} \\ & p^{val}_{k,n_3} & p^{val}_{k-1,n_3} & \ldots & p^{val}_{k-n,n_3} \\ & MA^{n_4}_k & MA^{n_4}_{k-1} & \ldots & MA^{n_4}_{k-n} \\ & RSI^{n_5}_k & RSI^{n_5}_{k-1} & \ldots & RSI^{n_5}_{k-n} \\ & WL^{n_6}_k & WL^{n_6}_{k-1} & \ldots & WL^{n_6}_{k-n} \\ & FI_k & FI_{k-1} & \ldots & FI_{k-n} \\ & vol_k & vol_{k-1} & \ldots & vol_{k-n} \\ & H^n_{r,k-n} & H^n_{r,k-n-1} & \ldots & H^n_{r,k-2n} \end{array} \tag{15}$$

$$\begin{array}{cccccc} y & \bar{x} & & & & \\ L^n_{rk} & r_k & r_{k-1} & \dots & r_{k-n} \\ & p^{val}_{k,n_1} & p^{val}_{k-1,n_1} & \dots & p^{val}_{k-n,n_1} \\ & p^{val}_{k,n_2} & p^{val}_{k-1,n_2} & \dots & p^{val}_{k-n,n_2} \\ & p^{val}_{k,n_3} & p^{val}_{k-1,n_3} & \dots & p^{val}_{k-n,n_3} \\ & MA^{n_4}_k & MA^{n_4}_{k-1} & \dots & MA^{n_4}_{k-n} \\ & RSI^{n_5}_k & RSI^{n_5}_{k-1} & \dots & RSI^{n_5}_{k-n} \\ & WL^{n_6}_k & WL^{n_6}_{k-1} & \dots & WL^{n_6}_{k-n} \\ & FI_k & FI_{k-1} & \dots & FI_{k-n} \\ & vol_k & vol_{k-1} & \dots & vol_{k-n} \\ & L^n_{r,k-n} & L^n_{r,k-n-1} & \dots & L^n_{r,k-2n} \end{array} \tag{16}$$

where r stands for the rate of return, p^{val} denotes p-value, MA is a moving average of the rate of return, RSI is the relative strength index, WL denotes the Williams oscillator, FI is the force index, and vol stands for the volume.

We assume the following values of the parameters representing various time horizons: $n_1 = 7$, $n_2 = 14$, $n_3 = 35$, which correspond to one day, 2 days and 5 days (i.e. a working week), respectively. On the other hand $n_4 = 21, n_5 = n_6 = 7$ are parameters often used for oscillators.

As it is a single snapshot window is described by 10 parameters. Since we take snapshots for $n = 28$ hours (i.e. 4 working days), the single input for the classifier is 280 dimensional.

2.4 Transactional Rule

Suppose that $\hat{L}^r_k$, $\hat{H}^r_k$ are the predicted low and high relative rates of return obtained from our model. The main concept of trading is to buy when prices are close to the low boundary and sell when the current price is close to the predicted maximum price value in next 28 hours. This idea might be described by the following algorithm:

if position==NONE AND $\hat{L}^r_k$ is in uptrend **then**
  if $\frac{|C_i - C_i \hat{L}^r_k|}{C_i} \leq \sigma$ **then**
    BUY
    buyPrice=Ci
    position=LONG
  end if
else
  if position==LONG AND $\hat{H}^r_k$ is in downtrend **then**
    if $\frac{|C_i - C_i \hat{H}^r_k|}{C_i} \leq \sigma$ **then**
      SELL
      sellPrice=Ci
      position=NONE
    end if

end if
end if
if position==LONG AND $\frac{buyPrice-Ci}{buyPrice} >= \theta$ **then**
SELL
sellPrice=Ci
position=NONE
end if

Here, constant C_i-denotes the close price. Parameters of the transactional rule are set to $\sigma = 0.05$ and $\theta = 0.07$. The last IF clause realizes the stop loss condition. A trend existence in predicted values $\hat{L}_k^r$ and $\hat{H}_k^r$ is tested by with test (3).

3 Walk Forward Testing

It is crucial for trading algorithms to backtest them in a proper way. In particular, it means that data used for training and testing should be separated. Testing on the sample which is used also for training may easily lead to the overfitting. It is very important to validate a model appropriately on the out-of-sample data.

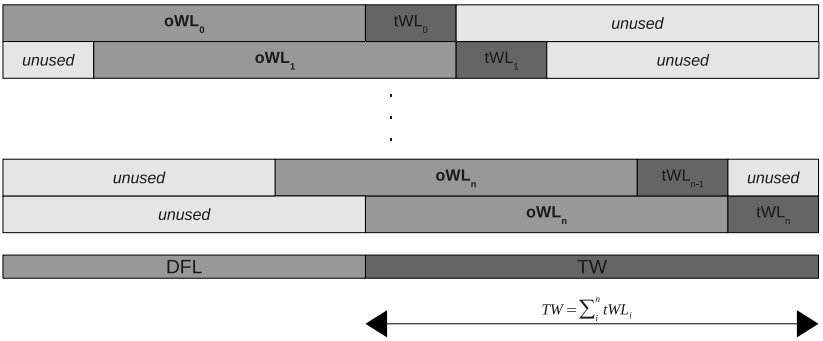


Fig. 1. Walk forward analysis

A classical approach for the model validation is to divide data into three parts used for training, testing and validation. However, this schema is not valid for the financial data where face non-stationary time series. For such class of problems the so-called walk forward analysis can be applied. Its goal is to minimize the curve fitting on the out-of-sample data by shifting the moving window. The window is divided into two subsets. First oWL examples are used for estimating the model. Next, the model is tested on the data consisted of tWL samples. After that, the moving window is shifted by the tWL samples and the next iteration begins. The whole process is illustrated in Figure 1, where DFL denotes the data feed length while TW means the testing window.

Our trading algorithm in it's base version did not manage to beat the market, although it adapted to the data generated from proposed mathematical model. Therefore we decided to proceed additional extensive test. Grid search optimization of free parameters σ, ρ, ϕ and θ were conducted to achieve better performance.

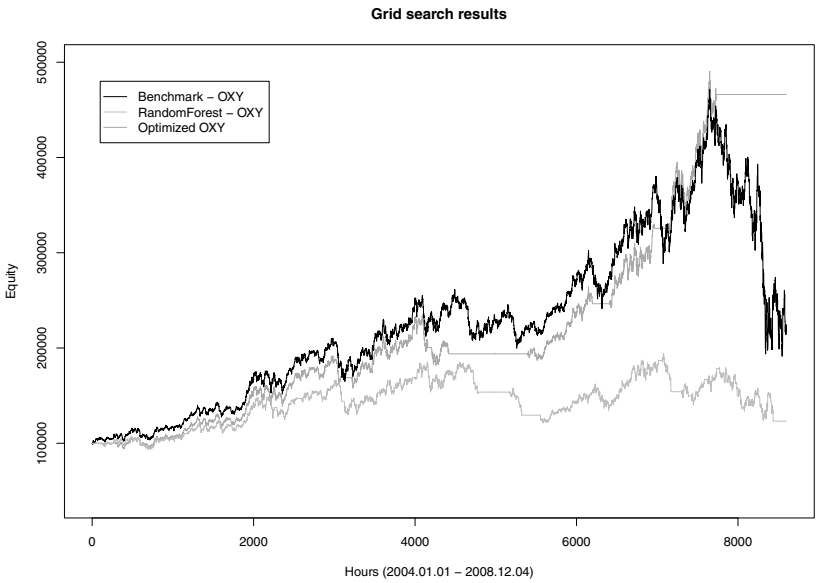


Fig. 2. Results of extensively optimized trading model

Results are shown in Figure 2. Algorithm earned 466119$ while benchmark ended up with nearly half of it. By applying the grid search model fitted the data and successfully followed the benchmark during bull market time. Moreover, it preserved earned money during consolidation phase and bear market times. For us the only conclusion that can be drawn from this experiment is that when someone search for too long, he will do data-snooping, like, on purpose, we did hear. With enough number of free parameters we can fit any data, especially using robust techniques, like one presented in this paper.

4 System Backtesting Results

The proposed system was verified by performing simulated trading of fifty randomly chosen shares from S&P500 index within dates 1'th of January 2004 to 8'th of August 2011. Average amount aggregates in single trajectory was equal to 150000. System was tested with walk forward methodology with $oWL = 1000$ and $tWL = 30$. Performing backtesting took 35 days of computations using 3-servers powerful cluster (see Fig. 3).

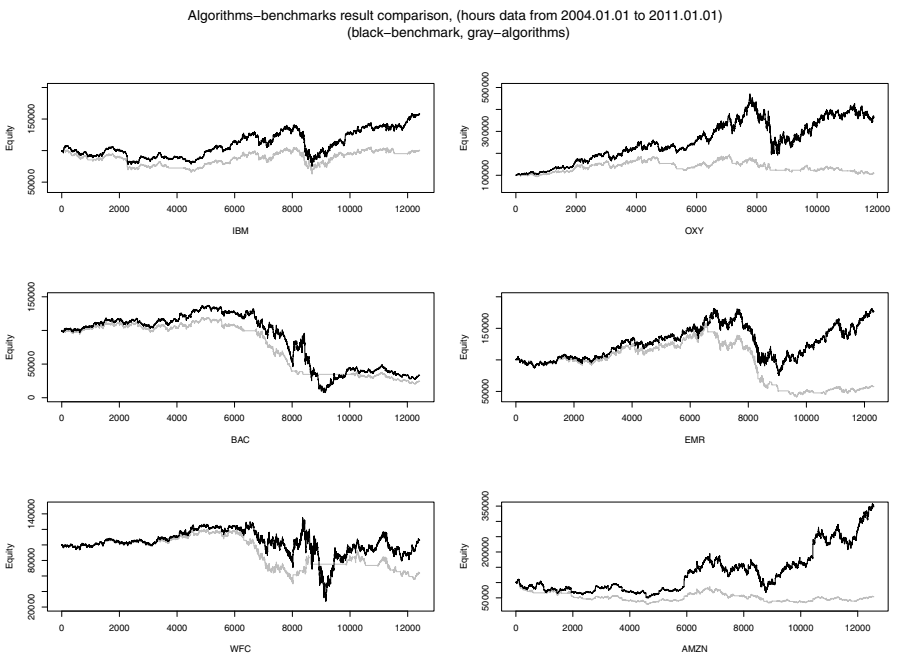


Fig. 3. Model performance trading sample shares from SP500 index

Fig. 4 shows accumulated from all trajectories weekly rate of returns histograms from benchmark and model. Expected value of week return from random share was equal to 0.002766 with corresponding weekly rate of return from model equal to −0.000597. As we can see from Fig. 3 and Fig. 4 buy and hold strategy turned out to be more profitable.

The total MSE of H^n_{ri} and L^n_{ri} prediction was equal to 0.019 0.021, respectively.

To check if the model have any predicting abilities we have constructed the following measures representing the average distance between the close price C_i at hour i and the maximum price in next 28 days, expressed in terms of the rate of return:

$$MSE_{(H^n_{ri},C)} = \sqrt{\sum_{i=28}^{N} \frac{1}{N-28} \left(\frac{max(C_{i-27}, C_{i-23}, \ldots, C_i) - C_{i-28}}{C_{i-28}} \right)^2} \quad (17)$$

$$MSE_{(L^n_{ri},C)} = \sqrt{\sum_{i=28}^{N} \frac{1}{N-28} \left(\frac{min(C_{i-27}, C_{i-23}, \ldots, C_i) - C_{i-28}}{C_{i-28}} \right)^2}. \quad (18)$$

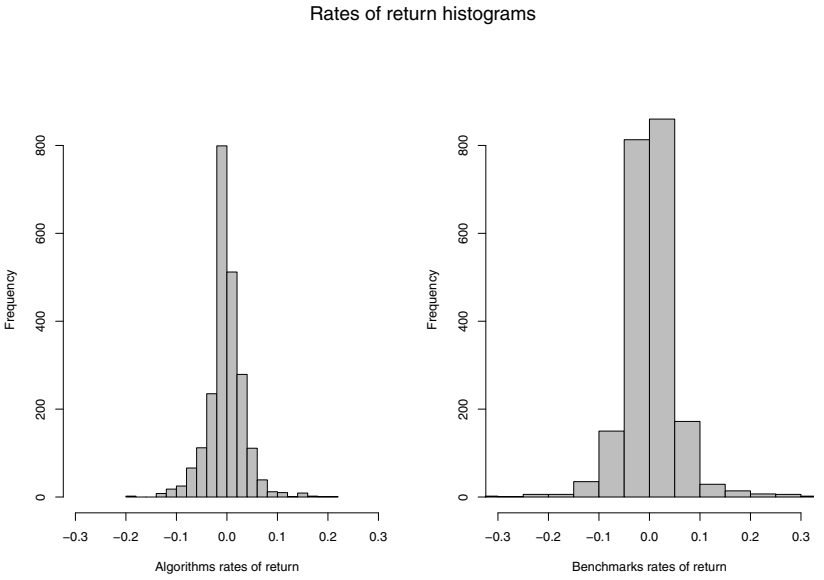


Fig. 4. Weekly rate of returns histograms from benchmark and model (50 randomly chosen shares from SP500)

If the model has no predicting power then these measures should have the same or greater rank than prediction MSE. We computed that $MSE_{(H^n_{ri},C)} = 0.024$, $MSE_{(L^n_{ri},C)} = 0.036$ what is slightly less than model MSE. It would be an interesting issue to formulate and verify precise statistical hypothesis if the model applied for price time series have any predicting power.

5 On-Line Adaptation to Non-stationary Two-Dimensional Mixed Black-Scholes Markov Time Series Model

Although the performance of the system on the real market is not satisfactory it is worth investigating its ability to adapt to non-stationary time series generated from a given model. Let $(S_i, Y_i)_{i=0}^n$ be a two dimensional time series. Let $(Y_i)_{i=0}^n$ be a discrete Markov process with the transition matrix

$$p(k,l) = P(Y_i = k | Y_{i-1} = l) = \begin{bmatrix} \frac{149}{150} & \frac{1}{150} \\ \frac{1}{70} & \frac{69}{70} \end{bmatrix}, \tag{19}$$

where $k, l = 0, 1$. Process Y has only two states and it simulates an important market indicator. Y proceeds serious price movements of the Black-Scholes trajectory S_i. $Y = 0$ - indicates an uptrend. If Y switches to 1, five hours later we have a 25% price downfall and uptrend turns into downtrend. On the other hand if Y switches from 1 to 0, then five hours later a downtrend turns into uptrend.

Let $S_0 = 100$ denote the initial price, $Y_0 = 0, a_0 = 1$ - initial trend (uptrend), $r = 0.005$, $\sigma = 0.002$. The following equations define dependent on Y_i price process S_i:

$$S_i = \begin{cases} S_{i-1} exp((a_i r - \frac{\sigma^2}{2}) + \sigma \mathbb{Z}_i), & \text{if } \mathbb{Y}_{i-5} = 0 \\ 0.75 S_{i-1}, & \text{if } \mathbb{Y}_{i-5} = 1 \end{cases} \tag{20}$$

$$a_i = \begin{cases} a_{i-1}, & \text{if } \mathbb{Y}_{i-5} = 0 \\ -a_{i-1}, & \text{if } \mathbb{Y}_{i-5} = 1. \end{cases} \tag{21}$$

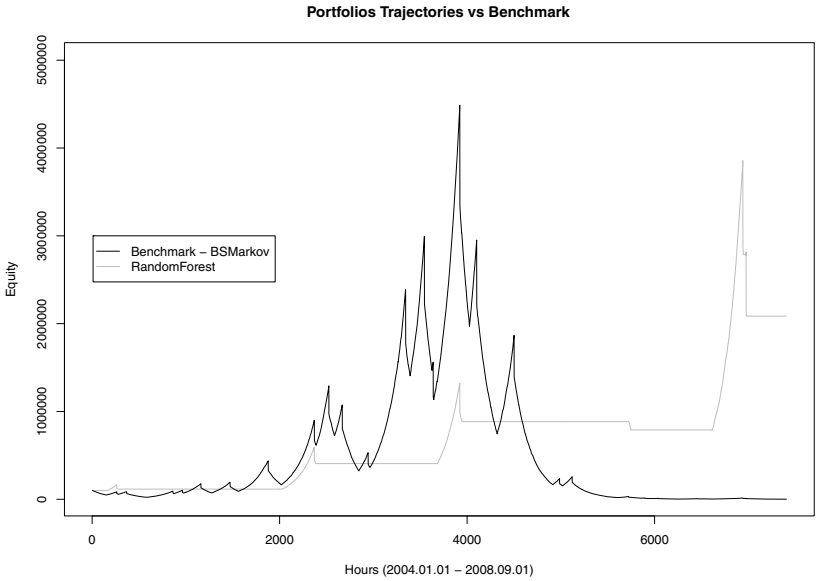


Fig. 5. Algorithm portfolio market value vs benchmark (generated model) portfolio market value

The simulation results are illustrated in Fig. 5. Variable Y models volume in the observation matrix for the classifier. MSE of predicted out-of-sample values $\tilde{H}^n_{ri}$ and $\tilde{L}^n_{ri}$ was equal 0.01 and 0.013 which are quite satisfactory results in comparison with the average four days (28 hours) high and low price movements given by (18) and (17), i.e. $MSE_{(H^n_{ri}, C)} = 0.08$, $MSE_{(L^n_{ri}, C)} = 0.11$.

Despite of good performance in terms of MSE, the system failed to deduce that state transitions of variable Y proceed dramatic price fall. Note that we have trained the forest in the window of 1000 hours length and the trend turns where rare phenomena during this period. It means that even a large but rare system prediction mistake has only a little impact on the regression absolute error measure of the random forest. Introducing some penalty factor for large prices downfall missclassifications seems to be a proper solution of this problem.

6 Conclusions

In this paper we presented a novel architecture of the system for automated stock trading and verified its performance on the large data set applying walk forward methodology. Although the system failed to generated profitable trading strategy it proved its ability of on-line adaptation to non-stationary time series. Hence we may conclude that application of machine learning techniques for on-line adaptation to real-life data models which distributions change drastically over time seems to be promising.

The major purpose of this work was to give an indication if machine learning techniques are able to generate profitable trading strategy. Although the trading results are not satisfactory there are some further issues worth examining. Firstly, one could try to incorporate information from other past prices and indicators. Secondly, it is worth of interest to examine other soft computing methods as a core of the system such as proposed in [6]. In particular, some clustering abilities of fuzzy sets may be useful for mining profitable trading rules.

References

1. Elder, A.: Trading for living. Wiley Finance (1993)
2. Chan, E.P.: Quantitive Trading. Wiley Trading (2008)
3. Fong, S., Si, Y.W., Tai, J.: Trend following algorithms in automated derivatives market trading. Expert Systems with Applications 39, 11378–11390 (2012)
4. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer (2009)
5. Jangmin, O., Jangwoo Lee, J., Jae Won, L., Byoung-Tak, Z.: Adaptive stock trading with dynamic asset allocation using reinforcement learning. Information Sciences 176, 2121–2147 (2006)
6. Kästner, M., Villmann, T.: Fuzzy Supervised Self-Organized Map for semi-supervised vector quantization. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part I. LNCS, vol. 7267, pp. 256–265. Springer, Heidelberg (2012)
7. Ładyżyński, P., Grzegorzewski, P.: Soft methods in trend detection. In: Borgelt, C., González-Rodríguez, G., Trutschnig, W., Lubiano, M.A., Gil, M.Á., Grzegorzewski, P., Hryniewicz, O. (eds.) Combining Soft Computing and Statistical Methods in Data Analysis. AISC, vol. 77, pp. 395–402. Springer, Heidelberg (2010)
8. Muh-Cherng, W., Sheng-Yu, L., Chia-Hsin, L.: An effective application of decision tree to stock trading. Expert Systems with Applications 31, 270–274 (2006)
9. Qing-Guo, W., Jin, L., Qin, Q., Shuzhi Sam, G.: Linear, Adaptive and Nonlinear Trading Models for Singapore Stock Market with Random Forests. In: 2011 9th IEEE International Conference on Control and Automation, ICCA (2011)

Proposal of an Inference Engine Architecture for Business Rules and Processes*

Grzegorz J. Nalepa, Krzysztof Kluza, and Krzysztof Kaczor

AGH University of Science and Technology
al. A. Mickiewicza 30, 30-059 Krakow, Poland
{gjn,kluza,kk}@agh.edu.pl

Abstract. In this paper, we discuss a new architecture for integrating and executing business process models with rules. It is based on a workflow engine that runs a BPMN-based business process model. On the lower level, rules are used to express the specific parts of the business logic. Rules working in the same context are grouped into a single task in the process model. Such a rule task is modeled by a formally defined decision table, which is designed in a visual way and its quality can be formally verified. In the runtime environment, tables are executed by a dedicated rule inference engine controlled by a workflow engine.

1 Introduction

Business rules are one of the preferred knowledge representation and processing solution in business applications. However, handling large rule bases remains a challenging task due to inference problems. Most of the existing inference engines use Rete-like inference which exploits a uninformed search strategy for rule selection in a given context. To solve this problem different inference control strategies are used. Number of business applications are designed with the use of business process (BP) models, especially modeled using Business Process Model and Notation (BPMN), that can express a high level control flow. Recently, the use of workflow engines that run the BP models to control rule engines has been advocated. However, the issue of providing a conceptually coherent architecture for integrating rules and processes remains unsolved.

In this paper, we discuss a new architecture for modeling, integrating and executing business process models with rules. This architecture is based on a workflow engine that runs a BPMN-based business process model. The engine is responsible for providing the inference control. On the lower level, business rules are used to express the specific parts of the business logic. Rules working in the same context are grouped into a single task in the business process model. Such a task is modeled by a formally defined decision table. On the runtime, tables are executed by a dedicated rule inference engine controlled by a workflow engine.

* The paper is supported by the HiBuProBuRul Project funded from NCN (National Science Centre) resources for science (no. DEC-2011/03/N/ST6/00909).

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 453–464, 2013.

Some related tools, such as Drools, offer a similar approach. However, the proposal presented here has important advantages. The rule model is defined in a formalized way which allows for the quality control of the rule base. Moreover, certain control parts of the BPMN process model can be transformed to rules.

The rest of this paper is organized as follows. In Section 2 the motivation for the paper is given with main problems outlined. Then in Section 3, the architecture of the proposed solution is described. The prototype implementation of this approach, composed of the jBPM process engine connected with the HeaRT rule engine is described in Section 4. In Section 5 the evaluation of this solution is given along with the discussion of related works. The paper is summarized with directions for future work in Section 6.

2 Motivation

Complex business intelligence applications require advanced modeling solutions. These solutions should be comprehensible for business architects, knowledge engineers and regular programmers. In the case of Business Process modeling, Unified Modeling Language (UML) is far too expressive and overcomplicated to be understood by an average business user [1]. Thus, the BPMN notation was introduced to deal with this issue. Although this notation can perfectly model a workflow in the process, the detailed logic of the process tasks can not be specified in pure BPMN.

The Business Rules (BR) approach [2,3], originated from the well-established Rule-Based Systems [4,5], constitute one of the solutions which can be used for the specification of task logic in Business Processes [6,7]. However, processes and rules are at different abstraction levels. Especially, rules in many systems are often specified in natural language, sometimes in a structured form e.g. SBVR (Semantics of Business Vocabulary and Business Rules) [8] or in an informal way. Such a difference between processes and rules poses both conceptual (in case of modeling) and practical (in implementation part) challenges, the so-called *semantic mismatch problem*. Therefore developing an integrated methodology and software for modeling BP with BR is extremely important in order to ensure the high quality of information systems in the future.

Although there is a difference in abstraction levels of BP and BR, rules can be complementary to processes. Such a separation of processes and rules is consistent with the BPMN 2.0 specification [9], which clarifies that BPMN is not suitable for modeling such concepts as rules. Thus, a BPMN process model should define the high level behavior of the system while the low level process logic can be described by rules.

Despite the ongoing research on such integration [6,10,11,12], there is no standardized and coherent methodology available. The integration details are not well specified, and there are not many tools which provide such an integrated modeling environment. Moreover, the existing approaches weakly support visualization of processes and rules, and they do not consider the quality issues.

A draft overview of a solution for integration of processes with rules we presented in [13]. Our solution supports integrating XTT2 BR with BPMN

processes. It uses advanced rule representations, such as decision tables or networks, for defining the task logic in the process. The rules in our representation are formalized using the Attributive Logic with Set Values over Finite Domains (ALSV(FD)) logic [14], and their syntax can be checked during the design process. As knowledge visualization helps in evaluation tasks [15], this solution takes advantage of visual aspects of the XTT2 knowledge representation.

The main issue considered in this paper is the architecture for execution of such an integrated model. It is based on combining a common process engine, with a custom rule engine. This solution addresses the implementation issues of the semantic mismatch problem.

This research is based upon our previous research [16] including visual inference specification methods for modularized rule bases [17] and selected analysis methods for Business Processes [18].

3 Architecture Proposal

The proposed integrated architecture for executing the combined BP and BR model is based on the open-source jBPM workflow engine that runs a BPMN-based BP model. The workflow engine communicates with the HeaRT rule engine that runs the low-level rule-based logic.

JBoss jBPM[1] is a workflow management system and a platform for executable process languages ranging from business process management over workflow to service orchestration. The system architecture is based on the WfMC's reference model [19]. jBPM provides a process engine for executing business processes and two optional core services (the history log for logging information about the current and previous state of all process instances and the human task service for the human task life cycle if human actors participate in the process). It has pluggable architecture and constitutes an extensible and customizable tool. Thus, it can be integrated with our runtime environment.

The tool allows for executing processes defined using the BPMN 2.0 XML format. However, the full BPMN 2.0 specification includes such details as choreographies and collaboration. The jBPM tool focuses only on specifying executable processes. It has been used for several implementation [20,21] and research [22,23,24,25] purposes concerning workflow modeling. In our research, only the subset of internal Business Process Model is considered.

In our solution, BPMN is used for modeling a workflow in the process, and the detailed logic of the process tasks is specified using the Business Rules approach, precisely – using the Semantic Knowledge Engineering (SKE) [26] approach.

In SKE, the eXtended Tabular Trees version 2 (XTT2) representation provides an expressive formal logical calculus for rules and a structured knowledge representation supporting the logical and visual specification of rules. XTT2 allows for advanced inference control [27] and formal analysis of the production systems [28]. It provides formal rule representation language [29], based on

[1] See: `http://www.jboss.org/jbpm/`

the ALSV(FD) logic [14]. Such knowledge representation makes the XTT2 language more expressive than propositional logic. Rules in the XTT2 rule base that work together in a common context are grouped into an extended decision table. An exemplary decision table for the CashPoint example, which determines the action of the ATM machine is presented in Figure 1. A header of the table contains attributes, which are common for all the rules in the table. Each rule, stored as a table row, consists of 5 condition columns marked as `(?)` (left-hand side part of a rule) and action column marked as `(->)` (right-hand side of a rule).

(?) authorized	(?) failedAttempts	(?) userRequestedAction	(?) udAmountDifference	(?) cdAmountDifference	(->) cashPointActivity
= false	< 3	= any	= any	= any	:= askForPIN
= false	= 3	= any	= any	= any	:= takeCardAway
= true	= any	= balance	= any	= any	:= displayBalance
= true	= any	= withdraw	>= 0	>= 0	:= payOut
= true	= any	= withdraw	= any	< 0	:= msgNotEnoughFoundsInMachine
= true	= any	= withdraw	< 0	>= 0	:= msgNotEnoughFoundsOnAccount

Table id: tab_2 - Table1

Fig. 1. An example of the XTT2 decision table

The model can be run in the HeKatE Run-Time Environment (HeART) rule engine [30]. However, HeART provides an executable environment for its internal representation in HMR.

In our solution, we used the native workflow engine for BPMN 2.0 – jBPM. It is a generic process engine based on the Process Virtual Machine (PVM)[2], which can execute business processes described in BPMN 2.0. We integrated the jBPM process engine with HeART.

An outline of the architecture for integration of jBPM with HeART is presented in Figure 2. Besides the discussed here runtime level, the whole framework addresses the aspects of modeling and verification of business process models with business rules. However, integration on modeling and verification level is out of scope for this paper.

An overview of the solution on the modeling level was given in [13]. It adapts the BPMN editor user interface for choosing a proper XTT2 decision table for a particular Business Rule task in the BPMN model.

During the design phase, XTT2 rules can be edited in the HQEd editor connected via network interface. In such a case opening a rule task in the BPMN editor allows for XTT2 editing in HQEd (see Figure 3). While executing, BR task triggers the selected XTT2 table.

The verification features of HeART can also be used for formal verification of selected BPMN models and rule tasks. Moreover, it is possible to verify the decision tables using HeART with the HeKatE Verification and Analysis (HalVA) framework [28]. However, this is mostly local verification of single rule tasks or simple BPMN constructs. A global verification of the complete model was considered in [18].

Thus, this general approach allows for visual modeling of processes and rules, their verification and execution of such a fully specified BPMN model.

[2] See: `http://docs.jboss.com/jbpm/pvm`

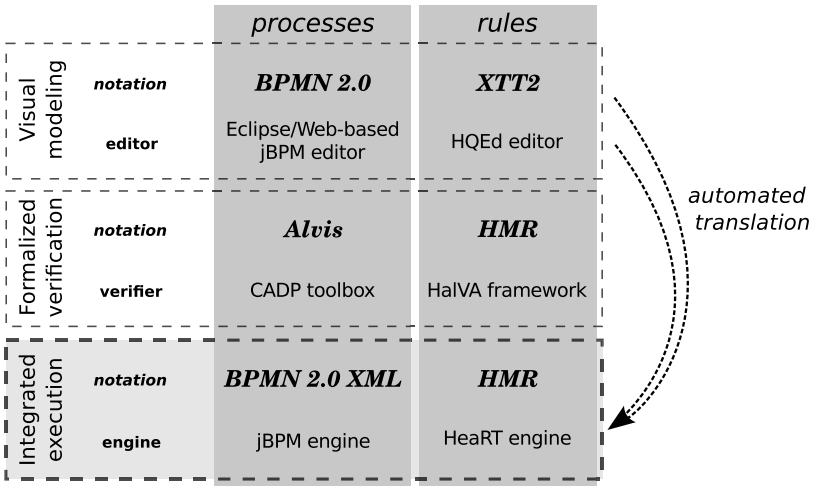


Fig. 2. An outline of the jBPM-HeaRT framework architecture

4 Prototype Implementation

A prototype implementation of the described architecture has been provided. The whole BP model is executed by jBPM that calls HeaRT for specific rule tasks using a socket-based interface. The HeaRT engine runs in a server mode and handles the communication from the jBPM run as a client. This is done by connecting the BPMN model tasks with the corresponding XTT2 tables.

In our solution, jBPM sends a message to HeaRT in the defined format:

1. *Rule task* with names starting from "H" are executed using HeaRT. Each rule task corresponds to one XTT2 table.
2. *Ruleflow groups* are associated with the list of XTT2 tables (optionally also with states) in the rule base.
3. If the model state is not specified, the HeaRT engine starts from its current state.
4. State attributes which are results of the HeaRT inference are stored as context variables of the process instances.

Communication between these tools is possible via dedicated API. HeaRT introduces an integration layer for the most common programming languages, including Java, PHP and Python. The integration module was designed to allow many clients to work with one HeaRT instance acting as an inference server. HeaRT communication with jBPM uses a network protocol and is handled by a dedicated `HConnect` class, which uses `JHeroic` library integration library. In turn, jBPM (Drools) provides a dedicated Drools Knowledge API[3]. This knowledge-centric API is written in Java and provides classes and interfaces for knowledge manipulation.

[3] See: `http://docs.jboss.org/jbpm/v5.1/javadocs`

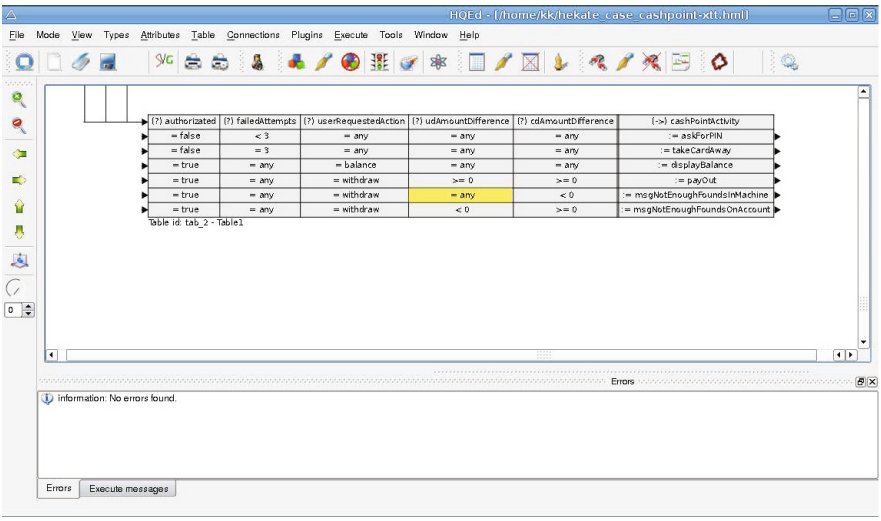


Fig. 3. An XTT2 decision table in HQEd Editor

The most common interfaces in the API are:

- `org.drools.builder.KnowledgeBuilder`
- `org.drools.KnowledgeBase`
- `org.drools.agent.KnowledgeAgent`
- `org.drools.runtime.StatefulKnowledgeSession`
- `org.drools.runtime.StatelessKnowledgeSession`

The communication with HeaRT is initialized by the jBPM engine during the execution of the process in Drools. The following code excerpt presents how the knowledge API and `HConnect` library can be used for executing a use case:

```
public static final void main(String[] args) {
     try {
         KnowledgeBase kbase = readKnowledgeBase();
         StatefulKnowledgeSession ksession =
            kbase.newStatefulKnowledgeSession();
         KnowledgeRuntimeLogger logger =
            KnowledgeRuntimeLoggerFactory.newConsoleLogger(ksession);
         //Read the HMR model from the filesystem and send it to HeaRT
         HConnect.addModel("CashPoint", new File("atm-rt.pl"));
         ksession.startProcess("HConnectTest");
         ksession.fireAllRules();
         logger.close();
     } catch (Throwable t) {
         t.printStackTrace();
     }
}
```

```
private static KnowledgeBase readKnowledgeBase() throws Exception {
     KnowledgeBuilder kbuilder =
         KnowledgeBuilderFactory.newKnowledgeBuilder();
     kbuilder.add(ResourceFactory.newClassPathResource("HConnectTest.bpmn"),
         ResourceType.BPMN2);
     KnowledgeBuilderErrors errors = kbuilder.getErrors();
     if (errors.size() > 0) {
         for (KnowledgeBuilderError error: errors) System.err.println(error);
         throw new IllegalArgumentException("Could not parse knowledge.");
     }
     KnowledgeBase kbase = KnowledgeBaseFactory.newKnowledgeBase();
     kbase.addKnowledgePackages(kbuilder.getKnowledgePackages());
     return kbase;
}
```

In this example, several elements can be observed: `KnowledgeBuilder` class for reading and compilation of the knowledge base from file, adding a new knowledge to knowledge base by using `KnowledgeBase` class, `HConnect` class for reading and adding XTT2 model to HeaRT, and executing model by calling `fireAllRules` method. The implementation was developed by Roman Sekuła [31].

As a proof of concept for the translation and integration of jBPM and XTT2, we modeled the CashPoint [32] example within this framework. CashPoint determines access to the customer's savings: a customer has to be authorized by inserting a card and typing in a correct PIN number, then the customer may make a cash withdrawal or ask for a balance to be printed. Figure 4 shows an example of the process model, which demonstrates the dependencies between rule tasks and XTT2 tables and the script task and the HeaRT output.

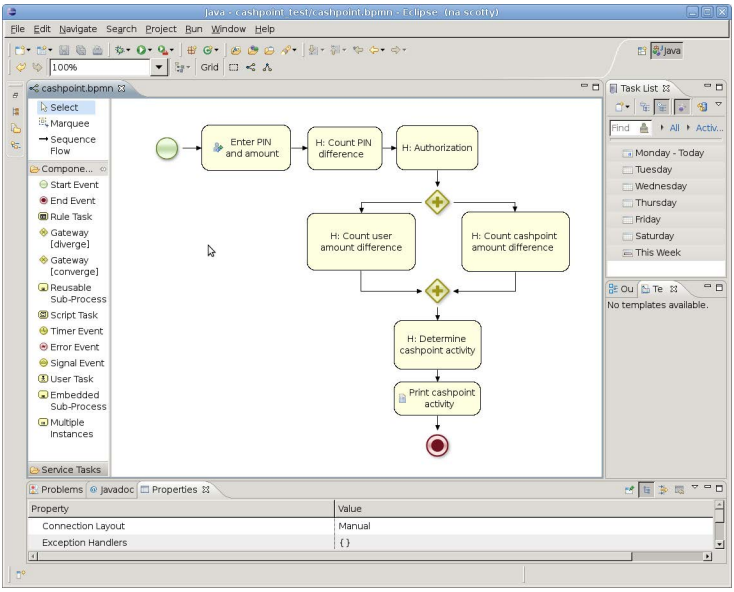


Fig. 4. An example of BPMN model for the CashPoint case study in jBPM

5 Evaluation and Related Work

The solution considered in this paper proposes a consistent approach, where business rules complement the high-level process model on a lower-level. The process model allows for business context identification. It also provides the general inference flow. Rules can be designed in a visual way. Moreover, the quality of the rule tasks can be verified formally e.g. w.r.t. rule completeness.

The most similar approach can be observed in the Drools framework. Originally the Drools rule engine used the so-called rule modules to group rules working in the same context. The rule grouping mechanism was used to limit the drawbacks of the Rete-like forward inference. Then, the Drools Flow engine was introduced. It offered a simple workflow-like mechanism to switch the predefined rule modules. Gradually, Flow was replaced by the full process engine – jBPM. However, the integration of the rule engine with the process engine did not address the issues of rule design and process design.

Currently, Drools and jBPM provide several different overlapping design tools for rules and processes, but no visual design tool for rules. Moreover, there is neither consistent design and development methodology nor a tool chain backing it. Finally, the issue of rule verification is not addressed.

There are also some other approaches which consider using processes and rules. However, they are less similar to our approach because they mostly do not use BPMN, do not provide any verification features, or do not support execution of the integrated model.

Worklets [10], proposed by Adams et al., uses rules stored in the RDR (Ripple Down Rules) sets. However, this solution, apart from binary trees, does not consider any other complex rule representation. Moreover, it does not use the BPMN notation. rBPMN [11] constitute one of few examples of coherent methodology for modeling processes integrated with rules. However, Milanovic et al. extended the standard of BPMN notation. Thus, their solution is not applicable for the existing BPMN models. Eijndhoven et al. [12] proposed a method which allows for application of production rules to several points in business processes. However, they do not used rules for specifying logic of tasks.

AgentWork [33], in turn, used rules for supporting workflow adaptation. Goedertier and Vanthienen [34] formally modeled business protocols using rules. Although these workflow systems provide verification features, they do not use BPMN notation and do not allow for modeling rule tasks.

Knolmayer et. al [6] described a rule-based method for modeling workflows. In this approach, the ECAA rules were used to specify business processes. Its strong limitation is focusing only on several workflow patterns. A hybrid composition approach consisting of the BPEL process and several well-modularized Business Rules was proposed by Charfi and Mezini in [7]. The approach consists of two phases which separate rules from processes. Another approach based on BPEL was proposed by Rosenberg and Dustdar in [35]. They described how to integrate business rules in a service-oriented way into BPEL. Although both of these approaches presents the integration method for BPEL processes and rules,

these approaches do not consider the BPMN standard and do not provide any specification of implementation.

Zur Muehlen et. al [36] considered the relationship between rules and processes. They analysed the representation capabilities of two rule modeling languages, SRML and SBVR, in comparison to the Petri net, EPC, IDEF and BPMN 1.0 approaches. In [37] zur Muehlen et. al compared BP-oriented and BR-oriented approaches and presented a decision framework for process and rule modeling. Some guidelines for process and rule modeling according to the particular factors, such as change frequency, implementation responsibility etc. were described. This should help modelers to decide how particular aspects of the organization should be modeled. However, the presented decision framework does not contain the integrated methodology.

Di Bona et al. [38] proposed using BPMN for modeling business rules. Although it is not a proper use of BPMN [9], such a solution can be useful for business analysts, which can have a clear graphical overview of the rule base.

The abovementioned papers do not provide any formalized specification of implementation. Several of them do not concern the BPMN notation, which is de facto the standard for process modeling. Moreover, most of these solutions are not supported by any tool chain.

There are several tools for process modeling which are partially adapted to process with rules modeling. However, there is no standardized and coherent methodology for their integration, and the existing solutions much differ in terms of aims and scope.

In Corel iGrafx Process[4] rules can be modeled using BPMN gateways, which controls the flow according to the value of attribute. However, this is very limited solution. It is not possible to group such defined rules in decision tables or reuse them, and iGrafx does not allow for using rules in BPMN tasks. IBM WebSphere Business Modeler Advanced[5], in turn, supports the Business Rule tasks in models. Such a task requires to define the `if-then` rule schema and to specify the rules. However, this solution does not provide a visual specification of rules. Business Process Visual Architect[6] allows a user to depict single rules and rule grids within a BPMN model. However, such rules are only depicted in the diagram and can not be used in simulation or execution process.

Our solution outlines a tool framework which supports integrating XTT2 business rules with BPMN processes. In contrast to the abovementioned tools which mostly do not support advanced rule representations, our solution uses decision tables for defining the task logic in the processes. Such tables can be visually designed using the HQEd editor. The rules in a table are formalized using the ALSV(FD) logic and their syntax can be checked during the design process. Moreover, our solution provides the integrated environment for executing of such modeled processes. A prototype implementation uses the jBPM process engine integrated with the HeaRT inference engine.

[4] See: `http://www.igrafx.com/`
[5] See: `http://www-01.ibm.com/software/integration/wbimodeler/`
[6] See: `http://www.visual-paradigm.com/product/bpva/`

6 Conclusion and Future Work

In the paper, we considered an architecture for execution of an integrated model combining with the SKE approach. This is a part of our research concerning integration of business processes with rules.

We proposed a framework architecture where specific tasks in a BPMN model are described using business rules specified with the XTT2 notation. XTT2 allows for modeling rules using decision tables described in a formalized way. Our framework allows for integrated executing of a specified model. As a proof of concept, we presented a prototype implementation, which uses the jBPM process engine integrated with the HeaRT inference engine for this purpose.

Several future directions for this research are considered. We plan to develop a consistent methodology which effectively uses the BPMN notation for modeling Business Processes with logic tasks defined using Business Rules based on the SKE approach. Such methodology by specifying the desirable design patterns for modeling processes with rules will eliminate a semantic mismatch problem. Another important issue is the quality of the integrated model. The verification features of HeaRT can be used for formal verification of selected BPMN models and rule tasks. However, this is mostly local verification of single rule tasks or simple BPMN constructs [39]. A global verification of the complete model, which uses the Alvis modeling language, was considered in [18].

References

1. Nalepa, G.J., Kluza, K.: UML representation for rule-based application models with XTT2-based business rules. International Journal of Software Engineering and Knowledge Engineering (IJSEKE) 22(4), 485–524 (2012)
2. Ross, R.G.: Principles of the Business Rule Approach, 1st edn. Addison-Wesley Professional (2003)
3. von Halle, B.: Business Rules Applied: Building Better Systems Using the Business Rules Approach. Wiley (2001)
4. Buchanan, B.G., Shortliffe, E.H. (eds.): Rule-Based Expert Systems. Addison-Wesley Publishing Company, Reading (1985)
5. Ligęza, A., Nalepa, G.J.: A study of methodological issues in design and development of rule-based systems: proposal of a new approach. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1(2), 117–137 (2011)
6. Knolmayer, G., Endl, R., Pfahrer, M.: Modeling processes and workflows by business rules. In: Business Process Management, Models, Techniques, and Empirical Studies, pp. 16–29. Springer-, London (2000)
7. Charfi, A., Mezini, M.: Hybrid web service composition: Business processes md„eet business rules. In: Proceedings of the 2nd International Conference on Service-Oriented Computing, ICSOC 2004, pp. 30–38. ACM, New York (2004)
8. OMG: Semantics of Business Vocabulary and Business Rules (SBVR). Technical Report dtc/06-03-02, Object Management Group (2006)
9. OMG: Business Process Model and Notation (BPMN): Version 2.0 specification. Technical Report formal/2011-01-03, Object Management Group (January 2011)

10. Adams, M., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Worklets: A service-oriented implementation of dynamic flexibility in workflows. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 291–308. Springer, Heidelberg (2006)
11. Milanovic, M., Gaševic, D.: Towards a language for rule-enhanced business process modeling. In: Proceedings of the 13th IEEE International Conference on Enterprise Distributed Object Computing, EDOC 2009, pp. 59–68. IEEE Press, Piscataway (2009)
12. van Eijndhoven, T., Iacob, M.E., Ponisio, M.: Achieving business process flexibility with business rules. In: Proceedings of the 12th International IEEE Enterprise Distributed Object Computing Conference, EDOC 2008, pp. 95–104 (September 2008)
13. Kluza, K., Kaczor, K., Nalepa, G.J.: Enriching business processes with rules using the oryx BPMN editor. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 573–581. Springer, Heidelberg (2012)
14. Nalepa, G.J., Ligęza, A.: HeKatE methodology, hybrid engineering of intelligent systems. International Journal of Applied Mathematics and Computer Science 20(1), 35–53 (2010)
15. Baumeister, J., Freiberg, M.: Knowledge visualization for evaluation tasks. Knowledge and Information Systems 29(2), 349–378 (2011)
16. Nalepa, G.J.: Proposal of business process and rules modeling with the XTT method. In: Negru, V. (ed.) Symbolic and Numeric Algorithms for Scientific Computing, SYNASC Ninth International Symposium, September 26-29, vol. 506, pp. 500–506. IEEE Computer Society, IEEE, CPS Conference Publishing Service, Los Alamitos, California (2007)
17. Kluza, K., Nalepa, G.J., Łysik, Ł.: Visual inference specification methods for modularized rulebases. Overview and integration proposal. In: Nalepa, G.J., Baumeister, J. (eds.) Proceedings of the 6th Workshop on Knowledge Engineering and Software Engineering (KESE6) at the 33rd German Conference on Artificial Intelligence, Karlsruhe, Germany, September 21, pp. 6–17 (2010)
18. Szpyrka, M., Nalepa, G.J., Ligęza, A., Kluza, K.: Proposal of formal verification of selected BPMN models with Alvis modeling language. In: Brazier, F.M., et al. (eds.) Intelligent Distributed Computing V. SCI, vol. 382, pp. 249–255. Springer, Heidelberg (2011)
19. Hollingsworth, D.: The workflow reference model. Issue 1.1 TC00-1003, Workflow Management Coalition (January 1995)
20. Hu, J., Zhao, Z., Lv, Z.: Implementation of process management and control based on jbpm4.4. In: 2011 Second International Conference on Networking and Distributed Computing (ICNDC), pp. 218–221 (September 2011)
21. Huang, Y.Y., Jiang, R., Li, H.: A reusable system architecture based on jbpm and its application. In: Zhang, Y. (ed.) Future Communication, Computing, Control and Management. LNEE, vol. 142, pp. 517–525. Springer, Heidelberg (2012)
22. Bing, H., Dan-Mei, X.: Research and design of document flow model based on JBPM workflow engine. In: Proceedings from International Forum on Computer Science-Technology and Applications, IFCSTA 2009, vol. 1, pp. 336–339 (December 2009)
23. Peng, L., Zhou, B.: Research on workflow patterns based on jBPM and jPDL. In: Proceedings from IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, PACIIA 2008, pp. 838–843. IEEE (December 2008)

24. Wohed, P., Russell, N., ter Hofstede, A.H., Andersson, B., van der Aalst, W.M.: Patterns-based evaluation of open source BPM systems: The cases of jBPM, OpenWFE, and Enhydra Shark. Information and Software Technology 51(8), 1187–1216 (2009)
25. Ostermayer, L., Seipel, D.: Knowledge engineering for business rules in prolog. In: Proceedings of the 26th Workshop on Logic Programming (WLP 2012), Bonn, Germany, September 24-25. CoRR Computing Research Repository (2012)
26. Nalepa, G.J.: Semantic Knowledge Engineering. A Rule-Based Approach. Wydawnictwa AGH, Kraków (2011)
27. Nalepa, G., Bobek, S., Ligęza, A., Kaczor, K.: Algorithms for rule inference in modularized rule bases. In: Bassiliades, N., Governatori, G., Paschke, A. (eds.) RuleML 2011 - Europe. LNCS, vol. 6826, pp. 305–312. Springer, Heidelberg (2011)
28. Nalepa, G.J., Bobek, S., Ligęza, A., Kaczor, K.: HalVA - rule analysis framework for XTT2 rules. In: Bassiliades, N., Governatori, G., Paschke, A. (eds.) RuleML 2011 - Europe. LNCS, vol. 6826, pp. 337–344. Springer, Heidelberg (2011)
29. Nalepa, G.J., Ligęza, A., Kaczor, K.: Formalization and modeling of rules using the XTT2 method. International Journal on Artificial Intelligence Tools 20(6), 1107–1125 (2011)
30. Nalepa, G.J.: Architecture of the heaRT hybrid rule engine. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 598–605. Springer, Heidelberg (2010)
31. Sekuła, R.: Review of selected workflow environments. Technical report, AGH UST, BSc Thesis, G.J.Nalepa, PhD Supervisor (2012)
32. Denvir, T., Oliveira, J., Plat, N.: The Cash-Point (ATM) 'Problem'. Formal Aspects of Computing 12(4), 211–215 (2000)
33. Müller, R., Greiner, U., Rahm, E.: Agent work: a workflow system supporting rule-based workflow adaptation. Data Knowl. Eng. 51(2), 223–256 (2004)
34. Goedertier, S., Vanthienen, J.: Business rules for compliant business process models. In: Abramowicz, W., Mayr, H.C. (eds.) BIS. LNI, vol. 85, pp. 558–572. GI (2006)
35. Rosenberg, F., Rosenberg, F., Dustdar, S., Dustdar, S.: Business rules integration in BPEL – a service-oriented approach. In: Proceedings of the 7th International IEEE Conference on E-Commerce Technology (CEC 2005), pp. 476–479 (July 2005)
36. zur Muehlen, M., Indulska, M., Kamp, G.: Business process and business rule modeling languages for compliance management: a representational analysis. In: Tutorials, Posters, Panels and Industrial Contributions at the 26th International Conference on Conceptual Modeling - Volume 83. ER '07, Darlinghurst, Australia, Australia, Australian Computer Society, Inc. (2007) 127–132
37. zur Muehlen, M., Indulska, M., Kittel, K.: Towards integrated modeling of business processes and business rules. In: 19th Australasian Conference on Information Systems ACIS 2008, Christchurch, New Zealand (December 2008)
38. Di Bona, D., Lo Re, G., Aiello, G., Tamburo, A., Alessi, M.: A methodology for graphical modeling of business rules. In: 5th UKSim European Symposium on Computer Modeling and Simulation (EMS) 2011, pp. 102–106 (November 2011)
39. Kluza, K., Maślanka, T., Nalepa, G.J., Ligęza, A.: Proposal of representing BPMN diagrams with XTT2-based business rules. In: Brazier, F.M. (ed.) et al., eds.: Intelligent Distributed Computing V. Proceedings of the 5th International Symposium on Intelligent Distributed Computing – IDC 2011, Delft, the Netherlands, Oct. 2011. Studies in Computational Intelligence, vol. 382, pp. 243–248. Springer, Heidelberg (2011)

Emergence of New Global Airline Networks and Distributing Loads in Star Airports

Hidefumi Sawai

Advanced ICT Research Institute,
National Institute of Information and Communications Technology
588-2, Iwaoka, Nishi, Kobe 651-2492 Japan
sawai@nict.go.jp

Abstract. We have developed a method of emerging a small-world network, which has the shortest average path length among complex networks known so far, in a self-organizing manner using an ACO (Ant-Colony Optimization)-inspired method. We call it an n-Star network. As one of the real-world applications, we showed the n-Star network could be applied to reorganizing a next generation global airline network, where several star nodes are assigned to the corresponding star cities selected from major cities in the world in advance, and we evaluated the performance of the network using several kinds of network parameters. This method is a hybrid method using a bottom-up and top-down approach. In this study, without selecting any star cities in advance, using a bottom-up method only based on city population, city ranking, and distance between cities, etc., we tried to emerge a self-organizing global airline network by connecting links between important cities. As a result, the latter n-Star network is formed which is different from the former n-Star network, however, it is expected that both n-Star networks will concentrate heavy loads on their respective star airports. We will verify the concentration of load on the star airports through a simulation experiment, and propose an effective method for distributing the load over the whole airline network.

Keywords: self-organization, global airline network, Ant-Colony Optimization, n-Star network, Small-World, complex network.

1 Introduction

We have proposed a self-organizing method from a random graph for emerging a novel Small-World network with the shortest average path length [3], which is different from conventional Small-World [1]. This method is inspired from Ant-Colony Optimization (ACO) [9], which seeks the shortest path based on ants' pheromone trails. The emerged network has fully connected multi-stars and their peripheral nodes which connect with all star nodes and adequately connect with each other. Fig.1 shows the n-Star networks with the number of star nodes n=1, 2, 3 and 4 at the center. From these regular structures, we can analyze several network parameters such as average degree $\langle k \rangle$, average path

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 465–476, 2013.

length L, clustering coefficient C and so on. These results suggest that a degree of Small-World (we call it "Small-Worldness") can be defined by a value $R \equiv L + p = L + \langle k \rangle / (N-1)$, where p is a link probability, and N is the number of nodes. Through a theoretical analysis and simulation experiments, the R value of n-Star networks is always constant 2 independently from the values of L and p (i.e., $\langle k \rangle$ and N). For other complex networks, for examples scale-free networks and random graphs, except a complete graph and a single-star network, the R values always become more than 2. Consequently, this R value can be regarded as a degree of "Small-Worldness".

The n-Star networks with above advantageous property can be used for several real-world applications, such as analysis of SNS (Social Networking Service) and logistics in airline networks. We have studied feasibility of the n-Star networks for reorganizing a next generation global airline network as one of the logistics fields [4]. Star nodes in an n-Star network can be assigned to some major airports, and peripheral nodes are assigned to other airports in the world, using the latitudes, longitudes, populations, city rankings and geodesic distance between their belonging cities. This mapping method can be regarded as a top-down approach. Simulation experiments showed that the n-Star networks could provide the means for airplanes and their passengers to travel with the shortest average distance and the least number of transits between any two cities, which will be greatly beneficial for them by saving mileage and travel time by 20 to 30 percent in the region of relatively small values of $\langle k \rangle$ ($\langle k \rangle$ corresponds to the number of average flight service from each airport) compared to a current airline network.

This paper will propose a different approach (i.e., a bottom-up approach) from a random graph to emerge a global airline network based on the n-Star network, and compare the characteristics of the two different approaches. Here, an issue will happen independently from the top-down and bottom-up approaches, because some loads will concentrate on star airports in the n-Star network. First, we will verify the phenomena for the concentration of loads on star airports through simulation experiments, and propose some effective methods for alleviating the loads on the star airports. These methods show the feasibility for applying the n-Star network to a next generation global airline network in the near future.

2 Application of n-Star Networks to a Next Generation Airline Network

To apply the n-Star network to a global airline network, we first define several parameters related to the cities in a world city rank list [7][8] as follows:

- Population of city i: $Po(i)$.
- Geodesic distance between city i and city j on the Earth: $d(i,j)$.
- Traffic between city i and city j: $t(i,j)$.
  $t(i,j)$ can be defined in proportion to the multiplication of $Po(i)$ and $Po(j)$, i.e., $t(i,j) \propto Po(i)\ Po(j)$.

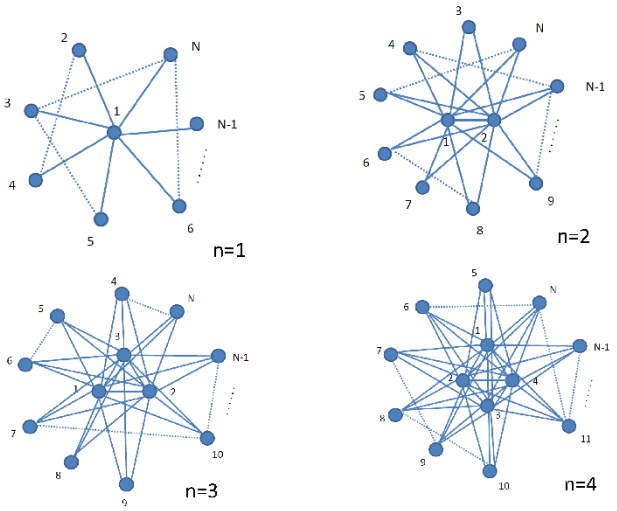


Fig. 1. Multi-star networks with star nodes at the center and their peripheral nodes. The star nodes fully connect with their peripheral nodes, and some peripheral nodes connect with other peripheral nodes, as denoted by dotted lines.

- Ranking of city i: $r(i)$.
- Weighted important relation (or value) between city i and city j: w_{ij}.

w_{ij} is defined as the value which is in proportion to city rankings $r(i)$ and $r(j)$, the traffic $t(i,j)$ and in inverse proportion to the distance $d(i,j)$, i.e., $w_{ij} \propto r(i)\ r(j)\ t(i,j)\ /\ d(i,j) \propto r(i)\ r(j)\ Po(i)\ Po(j)\ /\ d(i,j)$.

2.1 A Hybrid (Bottom-Up and Top-Down) Approach

An emerged n-Star network, which was self-organized by the ACO-inspired method, can be mapped onto a global airline network, as shown in Fig.2. This is a hybrid method for generating an airline network because a bottom-up approach and a top-down approach are combined with each other. As five star airports, Tokyo, Chicago, Frankfurt, Sydney and Johannesburg are selected from a world city rank list from a viewpoint of geographic and economic importance in each continent belonging to each city. As shown in this figure, the five star airports are fully connected with each other forming a *clique*, and almost all peripheral airports are connected with the star airports as long as the distances between them are within 90 percent of the cruising distance of an airplane (e.g., Boeing 787 [11] with the cruising distance 14,600 km). And many peripheral airports are connected with each other according to their weighted important relations or values, i.e., w_{ij}.

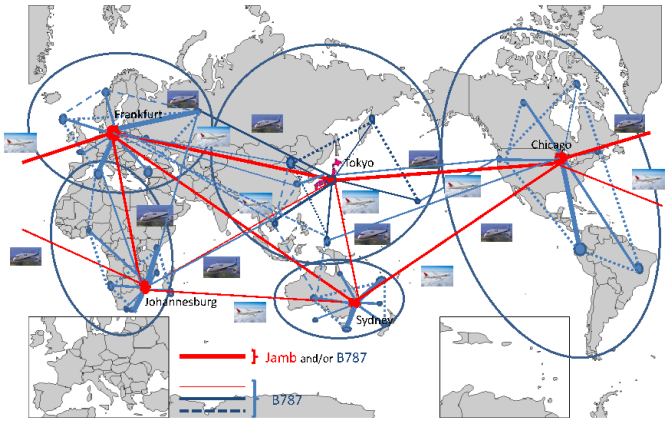


Fig. 2. Schematic diagram of a new generation global airline network using five star airports in an n-Star network. As the five star airports, Tokyo, Chicago, Frankfurt, Sydney and Johannesburg are selected from a viewpoint of geographic and economic importance in each belonging continent.

2.2 A Bottom-Up Approach

Starting from a random graph, an n-Star network can be emerged in a self-organizing manner. This is a bottom-up approach contrary to the former top-down/hybrid approach because no assignment of cities is arranged in advance. The procedure of forming an n-Star network is as follows:

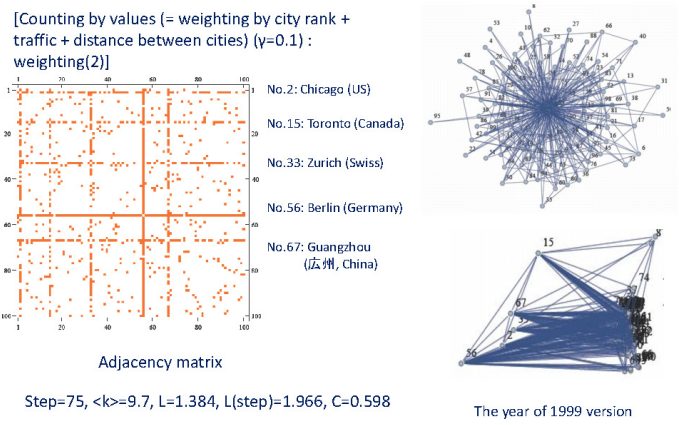


Fig. 3. An n-Star network emerged from a random graph in a bottom-up approach. As five star airports, Chicago, Toronto, Zurich, Berlin and Guangzhou are selected.

Each ant walks along the shortest path between any two cities/airports on an initial random graph, while spreading out pheromones with weighting the values w_{ij} between cities. The more pheromone spread on links, the thicker the concentration. In the next stage, all links with pheromones are sorted according to the order of pheromone concentration. Then both ends of the higher pheromone links are probabilistically exchanged with the next higher links (this operation increases the number of links and $\langle k \rangle$) while disconnecting the same number of lower pheromone links to keep the value of $\langle k \rangle$ constant. After reforming the network topology, the same procedure (i.e., sorting pheromone links, exchanging nodes and disconnecting links) is iterated until the average path length $L(dist)$ can reach under a predetermined value of average path length L_{th}. The values of $L(dist)$ (the shortest geodesic average distance between any two cities) and $L(step)$ (the shortest average steps between any two cities) monotonously decreased as the number of iterations increased.

Fig.3 shows an n-Star network emerged from a random graph in the bottom-up approach. As five star airports, Chicago, Toronto, Zurich, Berlin and Guangzhou are selected. This situation changes depending on an initial condition (an initial random graph) and a probabilistic procedure of exchanging pheromone links. In this case of network, the average degree $\langle k \rangle$ =9.7, $L(dist)$ = 1.34 (in the unit of the Earth's radius, about 6,360km), $L(step)$ = 1.96, clustering coefficient C = 0.598. This means that the average cruising distance between any two cities in the world city rank list is about 8,500km, and the average number of transits (this number is equals to $L(step)$-1.) is at most *once* because $L(step)$ is less than 2.

3 Phenomena of Load Concentration on Star Airports

It is unavoidable that a heavy load is concentrated on star airports if we apply an n-Star network to an airline network. However, first we should verify how much load concentration would occur at the star airports comparing to a current airline network with a scale-free characteristic modeled by a Barabási-Albert (BA) model [2]. We will discuss some effective methods to resolve this situation in the later section.

Here, we consider a situation that there are five star airports in an n-Star network such as Tokyo, Chicago, Frankfurt, Sydney and Johannesburg, as shown in Fig.2. We define $load(i)$ for city i as the summation of a frequency matrix $F(i,j)$ over all cities j (j =1,2, ...,N, $j \neq i$), where each component of the frequency matrix $F(i,j)$ is defined as a weighted ($\gamma = 0.01$) count of city value w_{ij} (i.e., $\gamma\ w_{ij}$) along the shortest paths between any two cities. The average load $\langle load \rangle$ over $N = 100$ cities is calculated by $\langle load \rangle = \sum_{i=1}^{N} load(i)/\mathrm{N}$.

Fig.4 shows the degree of load concentration at the five star airports without peripheral links, where the horizontal line itself represents the average load over 100 airports/cities. The vertical histogram is represented by Log_{10} scale.

The load for Frankfurt is over 10 times of the average load, and the loads of Tokyo, Chicago and Johannesburg are nearly ten times of the average load. The load of Sydney is much less than above loads. The load of most other cities are less than the average load.

Fig.5 shows the degree of load concentration at the five star airports with peripheral links. Compared to the case of Fig.4, the load concentration at the star airports are somewhat alleviated because of the peripheral links. However, there are still some airports at which the load concentration occurs more than ten times (Frankfurt) and almost ten times (Chicago and Johannesburg) of the average load.

3.1 Current Airline Networks

As a comparison to the n-Star network case, we analyze the load concentration of a current scale-free airline network, as shown in Fig.6. The network is often modeled by a BA (Barabási-Albert) model [2]. Fig.7 shows the distribution of loads in each airport. Although some loads concentrate at the top ranked (hub) airports, no extreme concentration at the star airports such as the n-Star network occurred and the loads are moderately distributed over all airports.

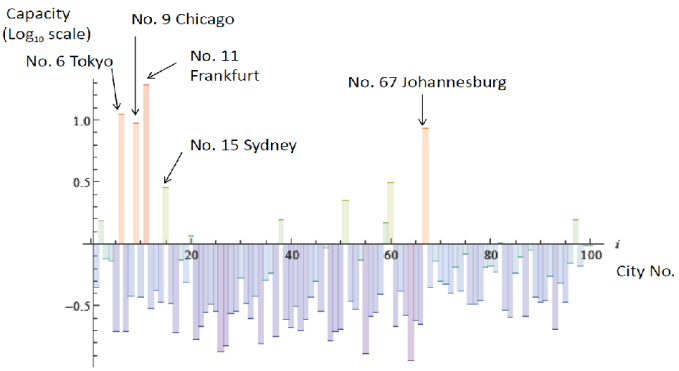


Fig. 4. Degree of load concentration in star airports without peripheral links. $\langle k \rangle$=8.36.

4 Distributing Loads in Major Airports

To alleviate the load concentration at star airports, there are mainly two methods: one method is to enlarge the capacity of star airports as much as possible, however, it might be difficult if some physical (i.e., spatial and temporal) capacity limitation exists on the airports. Another method is that multiple n-Star networks with a different set of star airports are prepared, and the load on star airports could be uniformly distributed over the multiple n-Star networks. We will consider the latter method in detail in the later section.

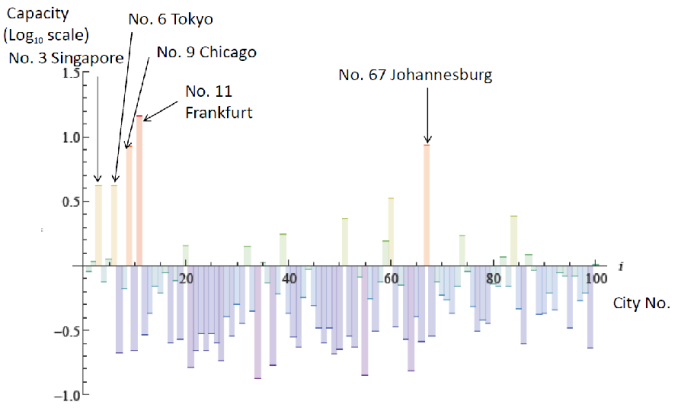


Fig. 5. Degree of load concentration in star airports with peripheral links. $\langle k \rangle$=16.3.

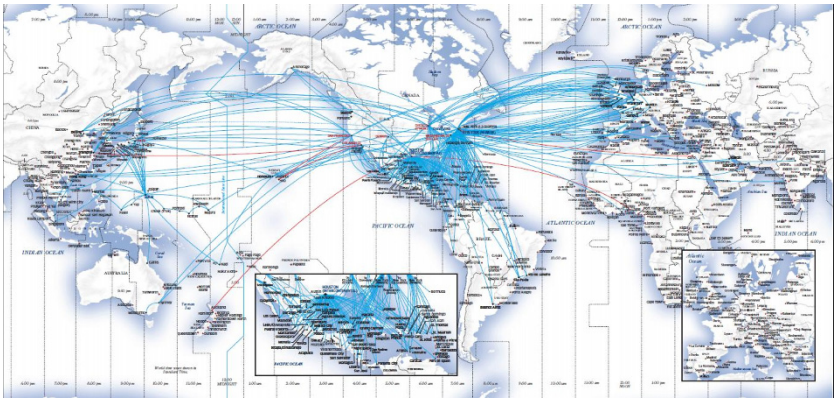


Fig. 6. A conventional global airline network [10]

4.1 Phenomena of Concentration of Loads on Major Airports in Three Groups

It is well known that there are three major airline groups in the world: Star Alliance, One World and Sky Team. Each group has each different belonging airline company with each representative airport. In Table 1, major airline companies and their stronghold airports are shown for each airline group. In this table, Tokyo is only an exception which belongs to two different airline groups, Star Alliance (ANA) and One World (JAL). Consequently, three kinds of n-Star networks with the different star airports can be formed using the stronghold airports in Table 1.

Load concentration on the stronghold airports in Star Alliance is shown in Fig.8 in the case of no peripheral links (i.e., no links represented by the dotted lines in Fig.1). In particular, the load concentrations on Frankfurt, Chicago and

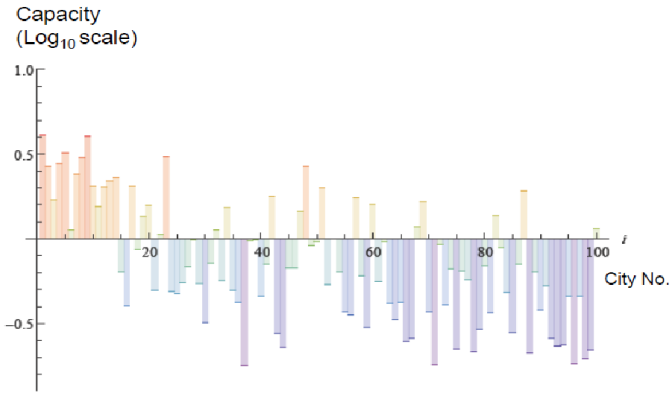


Fig. 7. Distribution of loads in each airport when a current (scale-free) airline network is modeled by a BA model. ($\langle k \rangle$=8.76).

Table 1. Three major airline groups and their belonging airline companies and strong airports

Group	Major Airline	Major airports
Star Alliance	Singapore Airlines, ANA, United Airline, Lufthansa, South African Airways	Singapore, Tokyo, Chicago, Frankfurt, Johannesburg
One World	American Airlines, British Airways, Cathay Pacific, JAL, Qantas	New York, London, Hong Kong, Tokyo, Sydney
Sky Team	KLM, Delta, Alitalia, Korean Air, Aeroflot,	Amsterdam, Atlanta, Roma, Incheon, Moscow

Johannesburg are found to be severe. The load concentration on stronghold airports in One World is shown in Fig.9 in the case of no peripheral links. Similar to the case of Star Alliance, the load concentration severely occurred at New York, London, Hong Kong and Tokyo. The load concentration at stronghold airports in Sky Team is shown in Fig.10. We found that the load concentration (as represented by the red bars) occurred at all of the stronghold airports in Sky Team.

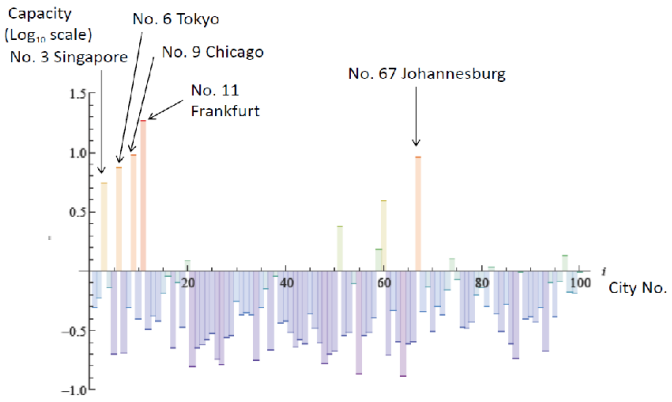


Fig. 8. Load concentration on stronghold airports in Star Alliance (Without peripheral links, $\langle k \rangle$=8.74)

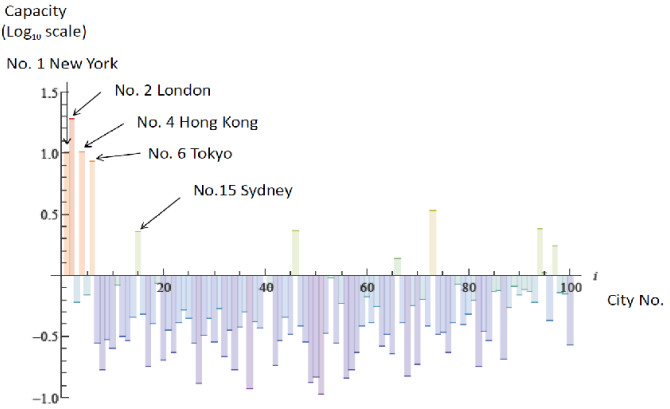


Fig. 9. Load concentration on stronghold airports in One World (Without peripheral links, $\langle k \rangle$=8.32)

To resolve and alleviate the load concentration on stronghold airports in each airline group, we use three different kinds of n-Star networks to distribute the loads in parallel over the whole global airline networks. In Fig.11, the distribution of loads is shown in the case of no peripheral links when distributing loads over the three airline groups. According to this figure, we found that the load in all stronghold airports of three airline groups becomes much less than ten times compared to the average load over the three airline networks (each is a different n-Star network). If there are peripheral links in the n-Star networks, as shown in Fig.12, the more load distribution occurred compared to the case of Fig.11 because the number of direct flights between peripheral airports increased, which will alleviate the load concentration on the stronghold airports.

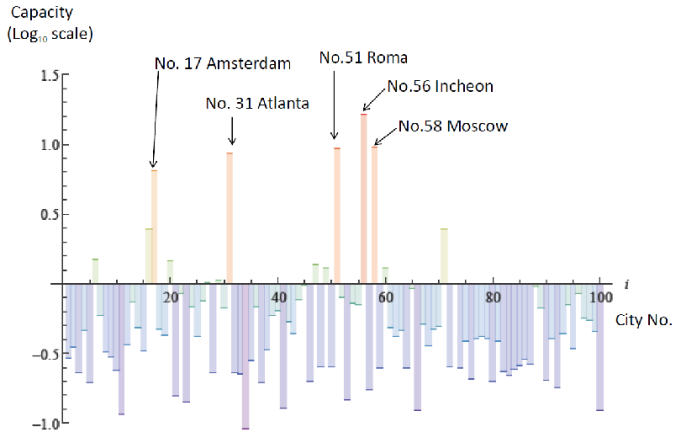


Fig. 10. Load concentration on stronghold airports in Sky Team (Without peripheral links, $\langle k \rangle$=9.16)

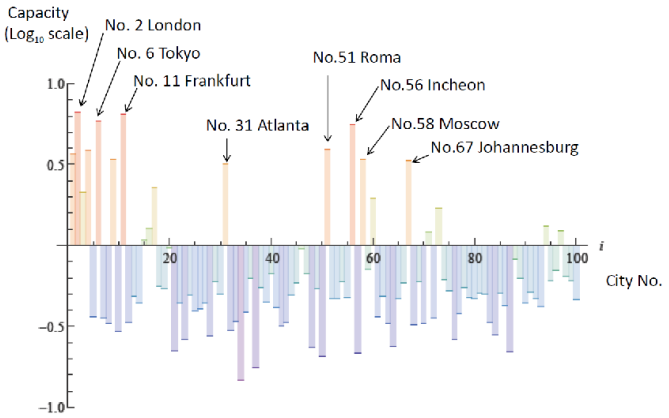


Fig. 11. Load distribution when distributing loads over the three airline groups. (Without peripheral links, $\langle k \rangle$=8.74).

5 Discussions

We describes the comparison of two approaches: the hybrid (combining bottom-up with top-down) and bottom-up approaches. As the bottom-up approach, which started from a random graph and performs a probabilistic process, sometimes two cities in the same continent are selected and no city is selected from one of five continents. For examples, no star city is sometimes selected from Australian continent, and two cities are selected from North American continent. On the other hand, in the hybrid (or top-down) approach there is the merit that

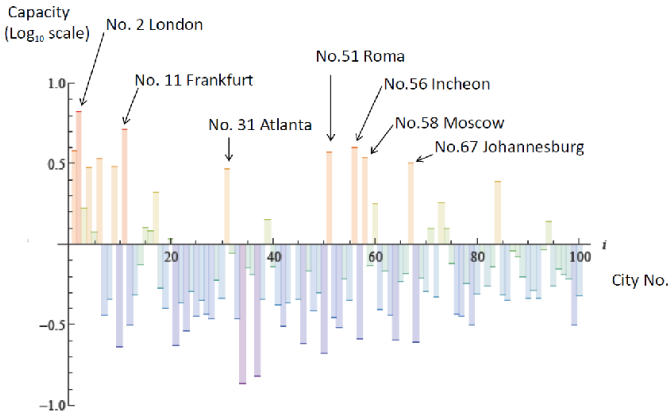


Fig. 12. Load distribution when distributing loads over the three airline groups. (With peripheral links, $\langle k \rangle$=16.3).

any star cities from each continent can be selected in advance from a geographic and economic point of view.

We found that the load concentration on major airports in n-Star networks could be controlled under at maximum 6.68 times of the average load over all airports if the load was distributed to three airline groups according to Figs. 11 and 12. This situation shows that the load is still concentrated on major airports compared to the current scale-free network in Fig.7, however, the amount of load will be able to dealt with the star airports in reality. If the load should be further alleviated, one possible way is for some worldwide operating airline companies (e.g., Emirates Airline with Dubai as the stronghold airport) to set another n-Star network different from those of the three airline groups, while separately dealing with the load within the capacity of major airports.

6 Conclusions

We described two methods how to apply an n-Star network to a next generation global airline network: one method is a hybrid (combining bottom-up with top-down) approach which assigns the star airports in the n-Star network to some predetermined major cities from five continents; another method is a bottom-up approach which emerges an n-Star network in a self-organizing manner from a random graph without assigning any star cities in advance. We compared their characteristics with each other. We verified the phenomena of load concentration on star airports, and discussed some effective and realistic methods to alleviate the load concentration by distributing it over different n-Star networks with different star airports. Some issues to be studied in the future include to construct several hierarchical structures [6] of airline network based on the n-Star network, and to verify the robustness [5] of the n-Star based airline network, because the airline routes are easily affected by the weather changes such as the eruption of volcanoes, typhoons, hurricanes, earthquakes, etc.

References

1. Watts, J.D.: Small Worlds: The Dynamics of Networks between Order and Randomness. Princeton University Press (1999)
2. Barabási, A.-L., Albert, R.: Emergence of Scaling in Random Networks. Science 286, 509–512 (1999)
3. Sawai, H.: Exploring A New Small-World Network for Real-World Applications. In: Proc. of CD-ROM, Int. Conf. on Networked Digital Technology 2012, Dubai, UAE, pp. 90–101 (April 2012)
4. Sawai, H.: Reorganizing A New Generation Airline Network Based on An Ant-Colony Optimization-Inspired Small-World Network. In: Proc. of the IEEE World Congress on Computational Intelligence 2012, Brisbane, Australia (June 2012)
5. Sawai, H.: A Small-World Network Robust for Both Random Failures and Targeted Attacks. In: Proc. of the 26th Annual Conference of the Japanese Society for Artificial Intelligence (June 2012) (in Japanese)
6. Sawai, H.: Structural Analysis on Hierarchical Multi-Star Small-World Networks. In: Proc., of the SIG-EIN (Special Interest Group on Emergent Intelligence on Network), JSSST (Japan Society for Software Science and Technology) (August 2012) (in Japanese)
7. Beaverstock, J.V., Smith, R.G., Taylor, P.J.: A Roster of World Cities 15(6),445–458 (December 1999)
8. Hot Spots, Benchmarking Global City Competitiveness, A Report from the Economist Intelligence Unit, Commissioned by Citi (January 2012)
9. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence, From Natural to Artificial Systems. Oxford University Press (1999)
10. `http://www.unitedairlines.co.jp/core/download/UAL_World_10_09.pdf`
11. Boeing 787 Dream Liner, `http://www.newairplane.com/`

Selection of Relevant Features for Text Classification with K-NN

Jerzy Balicki, Henryk Krawczyk, Łukasz Rymko, and Julian Szymański

Department of Computer Systems Architecture,
Gdańsk University of Technology, Poland
{jerzy.balicki,henryk.krawczyk,
julian.szymanski}@eti.pg.gda.pl,
lurym89@gmail.com

Abstract. In this paper, we describe five features selection techniques used for a text classification. An information gain, independent significance feature test, chi-squared test, odds ratio test, and frequency filtering have been compared according to the text benchmarks based on Wikipedia. For each method we present the results of classification quality obtained on the test datasets using K-NN based approach. A main advantage of evaluated approach is reducing the dimensionality of the vector space that allows to improve effectiveness of classification task. The information gain method, that obtained the best results, has been used for evaluation of features selection and classification scalability. We also provide the results indicating the feature selection is also useful for obtaining the commonsense features for describing natural-made categories.

Keywords: text representation, documents categorization, information retrieval, feature selection.

1 Introduction

A text document categorization algorithm can assign a document to some classes. Documents may be classified according to some features such as subjects, keywords, document type, author, printing year and so on. Classification automatically [1] improves effectiveness of data accessing and retrieval in textual repository.

Some vector space models are developed to represent a text and the dimensionality of such spaces grows larger with every distinct feature that may describe the document. Due to this property computing the similarity between articles become complex and a difficult task because high dimensional spaces causes effects of so-called *curse of dimensionality* [2]. In vector space models, depending on the metrics, all documents that are similar should be close due to distances between them. However most of the information used to create vector space models is not important to categorize thematic domain and may interfere into relationships between texts. Feature selection algorithms [3] [4] may help overcome this issue, by finding a subset of existing variables that are the most relevant to the training categories. This techniques can alleviate curse of dimensionality effect. Moreover, they can improve the classification to be much more computationally effective [5]. It is also frequent improvement of classification quality due to a fact that the feature reduction also may reduce the noise in the data.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 477–488, 2013.

This article describes some techniques of feature selection that were examined using real text datasets. Motivation of this work is to compare selected methods and then choose one that allow to achieve the best results. All of the methods are scalable and results may be computed faster with the usage of larger number of processing units.

2 Feature Selection Methods

In machine learning, a feature selection is a technique for choosing subset of features from vector space model in order to increase performance, minimize number of required operations or achieve better results [6]. For documents they may retrieve set of terms that are used more often in the particular group of articles. Some feature selection methods may belongs to the category of filter methods, which means that they treat every feature independently and apply single score to each of them. The selection is performed by removing all features that do not exceed chosen threshold or by choosing given number of features having the best scores. There are also wrapper methods [7], which use heuristics like greedy hill-climbing, simulated annealing or genetic algorithms. They usually choose result by iteratively evaluating candidate subset, modifying it and then deciding which one is better. These methods require scoring metric and they are computationally expensive, especially in high dimensional spaces and with large number of classes [8].

2.1 Notation

All feature selection methods described in the article use same notation:

- n_{kF} - number of documents that belong to the class k and contain feature F
- $n_{k\overline{F}}$ - number of documents that belong to the class k and do not contain feature F
- $n_{\overline{k}F}$ - number of documents that do not belong to the class k and contain feature F
- $n_{\overline{kF}}$ - number of documents that do not belong to the class k and do not contain feature F
- n_k - number of documents that belong to the class k
- $n_{\overline{k}}$ - number of documents that do not belong to the class k
- n_F - number of documents that contain feature F
- $n_{\overline{F}}$ - number of documents that do not contain feature F
- N - number of all documents

2.2 Information Gain Test

Information Gain [9] is one of the most popular feature selection method. This test evaluates how much information can be known about variable F, while knowing that variable A has value $A = a$. Score value of this test is the mutual information $I(F; A)$ which can be represented as reduction of entropy for variable F by information about value of variable A (represented by formula: $IG(F, a) = H(F) - H(F|a)$).

Let X represents the training set:

$$(\boldsymbol{x}, y) = (x_1, x_2, x_3, \ldots, x_n, y)$$

where $x_i \in vals(k)$ is a value of i-th attribute of testing vector $\boldsymbol{i}$ and y denotes the class label.

Information Gain score for training set X and for class k and entropy function $H()$ defines Formula 1.

$$IG(X,k) = H(X) - \sum_{v \in vals(i)} \frac{|\{\boldsymbol{x} \in X | x_i = v\}|}{|X|} \cdot H(\{\boldsymbol{x} \in X | x_i = v\}) \quad (1)$$

The sum of conditional entropies is equal to 0 when the every value v is unique for each of the classes (which means that the feature can be used to distinguish classes). Then the gain of information is equal to the entropy of $H(X)$, which is the greatest possible score.

Assuming that the set $vals(k)$ contains only two values (feature exist or not in the given document) and using the notation presented at the beginning of this section, the score for each class can be calculated using Formula 2.

$$IG(F) = -(\textstyle\sum_{k=1}^{C} \frac{n_k}{n_F} \log \frac{n_k}{n_F}) + \frac{n_F}{N}(\sum_{k=1}^{C} \frac{n_{kF}}{n_F} \log \frac{n_{kF}}{n_F}) + \frac{n_{\overline{F}}}{N}(\sum_{k=1}^{C} \frac{n_{k\overline{F}}}{n_{\overline{F}}} \log \frac{n_{k\overline{F}}}{n_{\overline{F}}}) \quad (2)$$

2.3 Independent Significance Feature Test (IndFeat)

Independent Significance Feature test [10] uses arithmetic mean and variance to evaluate feature score. This method assumes that whole set of observed objects belong to one of two classes. When more than two classes exist, then it is advised to compare classes in all possible pairs. When the score exceeds threshold at least for one pair, then it should not be removed. Equation 3 presents feature scoring used in this test.

$$IF(F) = \frac{|mean(A) - mean(B)|}{\sqrt{\frac{var(A)}{n_1} + \frac{var(B)}{n_2}}} \quad (3)$$

where A and B are variables that correspond to values of the feature F, but only for objects that belong to one of two compared classes. n_1 and n_2 are equal to the number of elements in classes.

In this work this method is improved. As training data have more than two classes feature scoring is calculated as the sum of all scores where single score is evaluated as IndFeat test of class A and all the other classes treated as class B. So the feature selection is performed according to Formula 4.

$$IF(F) = \frac{|C \cdot mean(A) - mean(B)|}{\sqrt{\frac{var(A)}{n_1} + \frac{var(B)}{n_2}}} \tag{4}$$

where C is a constant number bigger than 1.

Multiplying the mean of variable A makes it more important. Due to that change this test will choose positive features, which means that one that are distinctive for class A will get better score. It should be noticed the approach proposed here is similar to 1 vs all technique used to run binary classifiers working in multiclass mode.

2.4 Chi-squared Test χ^2

Chi-squared score ($\chi^2(F,k)$) measures the independence between feature F and class k. $\chi^2(F,k)$ equals zero when feature and class variables are independent.

The Formula 5 describes how Chi-squared test calculates feature scores.

$$\chi^2(F,k) = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \tag{5}$$

where r and c are the possible values of the feature and the class.

In case of text classification $r = c = 2$ because observation can contain feature F or not and observation may belong to class k or not. $E_{i,j}$ means theoretical frequency and $O_{i,j}$ is the observed frequency of existing feature i in category j. Using notation given in section 2.1 we can define observed frequency for each feature and class that is described in Tables 1 and 2.

Table 1. Table of observed frequency for each feature and class

$O_{i,j}$	F	$\overline{F}$	sum
k	n_{kF}	$n_{k\overline{F}}$	n_k
$\overline{k}$	$n_{\overline{k}F}$	$n_{\overline{kF}}$	$n_{\overline{k}}$
sum	n_F	$n_{\overline{F}}$	N

After noticing that when using values from the tables into the numerator of the fraction Equation (5) can be simplified to the Formula 6.

$$(O_{1,1} - E_{1,1})^2 = (n_{kF} - \frac{n_F \cdot n_k}{N})^2 = (\frac{n_{kF}(n_{kF} + n_{\overline{k}F} + n_{k\overline{F}} + n_{\overline{kF}})}{N} - \frac{(n_{kF} + n_{\overline{k}F})(n_{kF} + n_{k\overline{F}})}{N})^2 = (\frac{n_{kF}n_{\overline{kF}} - n_{\overline{k}F}n_{k\overline{F}}}{N})^2 \tag{6}$$

Table 2. Table of theoretical frequency for each feature and class

$E_{i,j}$	F	$\overline{F}$	sum
k	$\frac{n_F \cdot n_k}{N}$	$\frac{n_{\overline{F}} \cdot n_k}{N}$	n_k
$\overline{k}$	$\frac{n_F \cdot n_{\overline{k}}}{N}$	$\frac{n_{\overline{F}} \cdot n_{\overline{k}}}{N}$	$n_{\overline{k}}$
sum	n_F	$n_{\overline{F}}$	N

Analogous operations can be performed for every other values in the table and they are always equal:

$$(O_{1,1} - E_{1,1})^2 = (\frac{n_{kF}n_{\overline{kF}} - n_{\overline{k}F}n_{k\overline{F}}}{N})^2 = (O_{1,2} - E_{1,2})^2 = (O_{2,1} - E_{2,1})^2 = (O_{2,2} - E_{2,2})^2 \quad (7)$$

Using the Equation (7) whole Chi-squared score can be simplified:

$$\chi^2(F,k) = \sum_{i=1}^{r}\sum_{j=1}^{c} \frac{(\frac{n_{kF}n_{\overline{kF}} - n_{\overline{k}F}n_{k\overline{F}}}{N})^2}{E_{i,j}} =$$

$$(\frac{n_{kF}n_{\overline{kF}} - n_{\overline{k}F}n_{k\overline{F}}}{N})^2(\frac{N(n_k n_F + n_{\overline{k}}n_F + n_k n_{\overline{F}} + n_{\overline{F}}n_{\overline{k}})}{n_k n_{\overline{k}} n_F n_{\overline{F}}}) =$$

$$(\frac{n_{kF}n_{\overline{kF}} - n_{\overline{k}F}n_{k\overline{F}}}{N})^2(\frac{N(n_k(n_F + n_{\overline{F}}) + n_{\overline{k}}(n_F + n_{\overline{F}}))}{n_k n_{\overline{k}} n_F n_{\overline{F}}}) =$$

$$(\frac{n_{kF}n_{\overline{kF}} - n_{\overline{k}F}n_{k\overline{F}}}{N})^2(\frac{N^3}{n_k n_{\overline{k}} n_F n_{\overline{F}}}) \quad (8)$$

Final formula for scoring the feature F for class k is described with the Equation 9.

$$\chi^2(F,k) = \frac{N(n_{kF}n_{\overline{kF}} - n_{\overline{k}F}n_{k\overline{F}})^2}{n_k n_F n_{\overline{k}} n_{\overline{F}}} \quad (9)$$

Value of the feature can be evaluated as the weighted mean value of the Chi-squared scores for all classes (where weights are the probabilities of the class occurrence) that describes Formula 10.

$$\chi^2(F) = \sum_{k=1}^{C} Pr(y = k)\chi^2(F,k) \quad (10)$$

2.5 Odds Ratio Test

Odds Ratio score measures independence between two random events. Odds is the ratio of probability, that the event will occur to the probability that the event will not occur.

Score is evaluated as the ratio of the odds that feature F will be present in document from class k to the odds that feature F will not be present in document from class k.

Table 3 describes the probabilities of the events.

Table 3. Table of the conditional probabilities of existing feature F

	F	$\overline{F}$
k	$P(F\|k) = \frac{n_{kF}}{n_k}$	$P(\overline{F}\|k) = \frac{n_{k\overline{F}}}{n_k}$
$\overline{k}$	$P(F\|\overline{k}) = \frac{n_{\overline{k}F}}{n_{\overline{k}}}$	$P(\overline{F}\|\overline{k}) = \frac{n_{\overline{k}\overline{F}}}{n_{\overline{k}}}$

Using the definition of the Odds Ratio score and the probabilities from the Table 3 we obtain the Formula 11.

$$OR(F,k) = \frac{\frac{n_{kF}}{n_k} / \frac{n_{k\overline{F}}}{n_k}}{\frac{n_{\overline{k}F}}{n_{\overline{k}}} / \frac{n_{\overline{k}\overline{F}}}{n_{\overline{k}}}} = \frac{n_{kF} n_{\overline{kF}}}{n_{\overline{k}F} n_{k\overline{F}}} \approx \frac{(n_{kF}+C)(n_{\overline{kF}}+C)}{(n_{\overline{k}F}+C)(n_{k\overline{F}}+C)} \quad (11)$$

where C is a very small constant (e.g. 0.1) used to eliminate problems with dividing by 0.

Final score for the feature is evaluated as the weighted mean of the values for all classes that defines Formula 12.

$$OR(F) = \sum_{k=1}^{C} \frac{n_k}{N} \frac{(n_{kF}+C)(n_{\overline{kF}}+C)}{(n_{\overline{k}F}+C)(n_{k\overline{F}}+C)} \quad (12)$$

2.6 Frequency Filtering

Frequency Filtering score is the value of the number of documents that contain feature F in the class k, $FF(F,k) = n_{kF}$

Score for the feature F is the weighted mean for all classes described with Formula 13.

$$FF(F) = \sum_{k=1}^{C} Pr(y=k) n_{kF} = \sum_{k=1}^{C} \frac{n_k}{N} n_{kF} \quad (13)$$

3 Evaluation of Feature Selection Methods

The goal of experiments presented in this article is to evaluate how particular feature selection method impacts text classification results. Another purpose is to check how much number of features affects classification performance.

For each of the feature selection techniques we perform test on solid input data created from Wikipedia database. This wide repository contains different kinds of articles from various fields of knowledge described in natural language. Beside text content it provides relationships between articles and their assignments to categories where it belongs to. All this information has been used to create evaluation datasets.

Evaluation of single feature selection method test is executed in a following steps:

- Representing all articles as a vector in vector space model
  First step is about representing text article as a single vector in vector space model. In this work the task is done using two different approaches.
  First one defines text feature as a distinct word. This means that every word in language implies additional dimension in space model. Words are usually abbreviated by removing prefixes and suffixes that do not change the meaning of the word. In this case feature selection method choose words that usually appear in texts from chosen category.
  Second method takes links between Wikipedia articles as a feature. In first case number of features is equal to the number of words in the dictionary, in second it is the number of existing articles.
- Selecting subset of features
  Feature selection methods are described in Section 2.
- Performing text classification using only selected features
  Text classification is a task that matches one article to one or more categories. Although there is plenty of algorithms that solve this task, the simplest one was chosen called K-Nearest Neighbour algorithm [11]. Object is classified to the class by majority of votes of its K nearest neighbours. Nearest neighbours are found using the distances (e.g. cosine distance) in the vector space model.
- Comparing results with actual category of the text
  Results can be analyzed using various methods. Most intuitively one should compare accuracies of the classification results but it can sometimes bias the results. It is better to compare precision or recall of the results but it depends on the application. Most universal relevance score is the F-measure (or F-score) which is harmonic mean of both precision and recall.

4 Datasets and Results

4.1 Datasets

For experiments we create datasets using Matrix'u application that allows to change Wikipedia content into computationable form [12]. All of them are based on the Simple English Wikipedia.

Three datasets used in experiments:

- GR
  This dataset contains articles that belong to two categories (Geography or Religion). Those categories can be easily distinguished.

Table 4. Sizes of the used datasets

Dataset Name	# of articles	# of categories	# of features (words)	# of features (links)
GR	35 983	2	89 837	49 262
8K	56 564	8	114 628	58 855
Science	51 085	29	116 681	57 333

– 8K
This dataset contains 8 biggest categories in Simple English Wikipedia (Everyday life, People, Geography, History, Religion, Science, Literature, Knowledge). It contains all articles of this database.
– Science
This dataset contains 29 categories that belong to the category of Science (Physics, Technology, Mathematics, Biology, Chemistry etc).

4.2 Classification Results

Tests were executed with one parameter that defines the number of features that are chosen during feature selection. During the tests number of chosen features varied from 100 to 30 thousands. Classification on full dataset was also performed and results have been given as a reference values. The achieved results for a particular representation method and dataset have been presented in Figures 1 - 6.

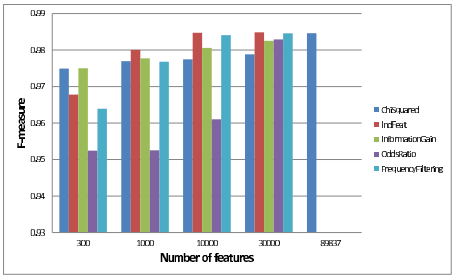


Fig. 1. Results for GR dataset with words feature representation

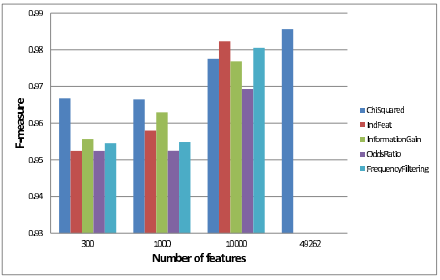


Fig. 2. Results for GR dataset with links feature representation

4.3 Common-Sense Features Interpretation

Table 5 presents some of the features that achieved best score using different selection methods. Tests were performed on GR dataset with word representation that contains only two categories: Geography and Religion. Selected features are generally consistent with the human intuition. Most commonly chosen features are for example: *god, religion, citi, region* or *unit*. Worth mentioning is the fact that different algorithms return different results and still most of them are able to achieve similar classification results e.g. Chi Squared and Information Gain favor features connected to religion (*god, belief,*

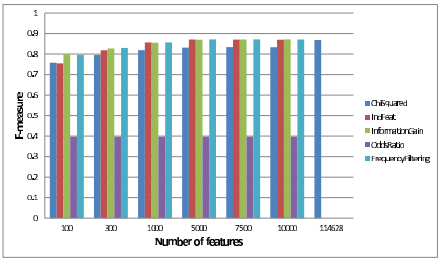


Fig. 3. Results for 8K dataset with words feature representation

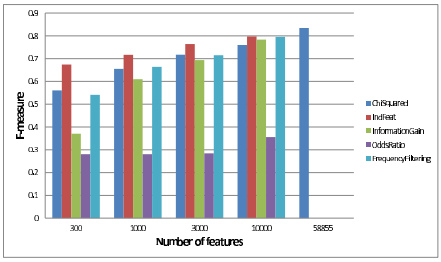


Fig. 4. Results for 8K dataset with links feature representation

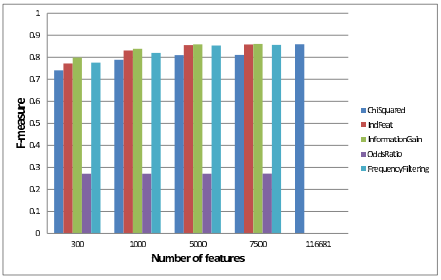


Fig. 5. Results for Science dataset with words feature representation

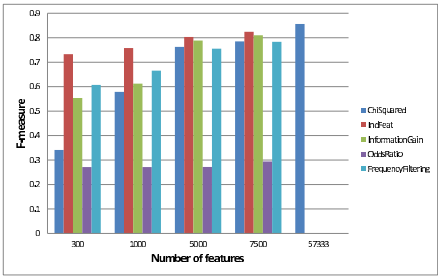


Fig. 6. Results for Science dataset with links feature representation

Table 5. Features that achieved greatest scores for different features selection methods (GR dataset)

Chi Squared	InfGain	OddsRatio	FrequencyFiltering	IndFeat
god	god	fullnam	name	commun
given	christian	dateofbirth	infobox	region
found	mytholog	countryofbirth	other	franc
book	believ	playernam	websit	depart
stori	bibl	cityofbirth	state	citi
refer	church	currentclub	citi	state
mean	greek	youthclub	found	found

religion, christian) where IndFeat returns features associated with geography (*commun, region, franc, citi*). Frequency Filtering and Odds Ratio methods returned words not necessarily intuitive (like *name* or *infobox*) and as it visible in figure 1 they return worse results. The results indicate the features selection can be used for refine the vector space and extract common-sense knowledge [13].

4.4 Scalability

The results of classification indicates that the one of the best approaches to feature selection is IndFeat. Usability of this method depends on the size of the dataset it was be applied to. We perform series of the experiments aiming at evaluation of scalability of proposed method. Scalability tests were performed on the 8K dataset.

In Figure 7 we provide time results for feature selection using IndFeat method in different number of processors used. What can be seen from the graph is that the method scales well – the time used for computations decreases in the function of processors.

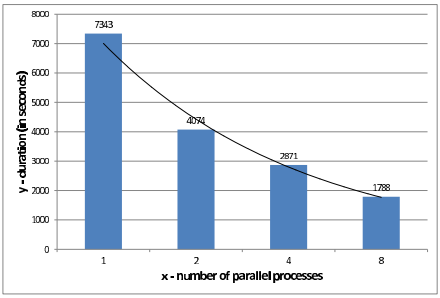


Fig. 7. IndFeat feature selection time for different number of parallel processes used

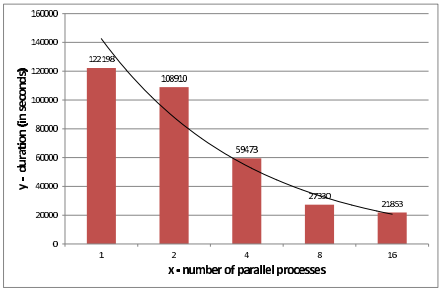


Fig. 8. K-NN classification time for different number of parallel processes used

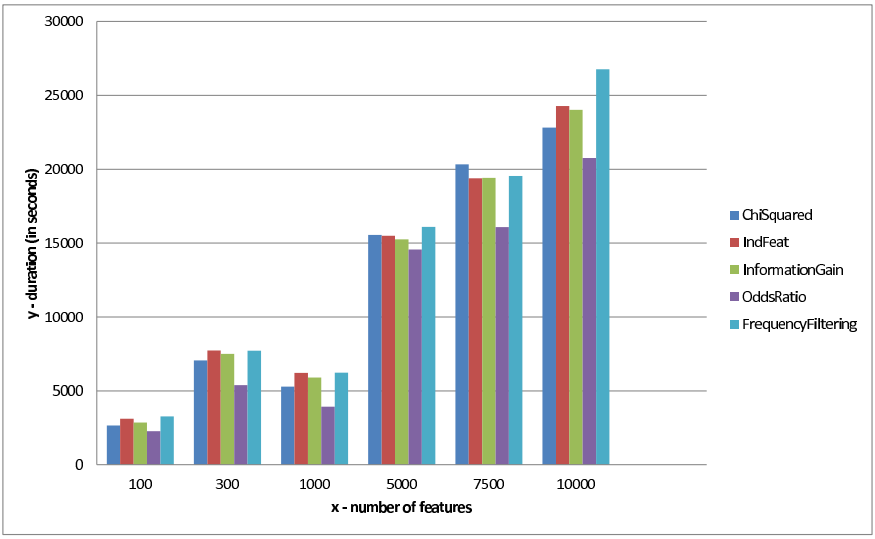


Fig. 9. Classification time with K-NN for different methods and different number of selected features

Considering categorization problems it is also important to evaluate how particular model scales-up. In Figure 8 we provide time results of K-NN classification on the 8K dataset. Also in that case graphs indicates the method scales-up well.

The time used for classification should be dependent on number features used for description objects in vector space. In Figure 9 we provide the times used for computations while different sizes of the features spaces were used. What is interesting the different selection methods for the same dataset and same number of features produce different time results. It is caused by the fact the different features have different density of values that relate them to the objects.

5 Conclusions

Feature selection methods are an ongoing research problem aiming to reduce the number of features from large datasets. The goal is to build better representation that reduce a noise in the data and also achieve the machine learning results faster. In the domain of text classification it is also used for discovering the dependencies between words and categories.

Test results presented above prove that it is possible to achieve satisfying time complexity and feature selection on big dataset. Removing more than 90% of the features does not distort the classification results too much.

The manual evaluation of particular features selected with proposed methods indicates the large number of them can be used as characteristic keywords that can distinguish categories between each other.

The results show that feature selection algorithms are able to reduce the time of the text classification. Comparison of five tested methods show that there is no single algorithm that can create the best feature set. Good scalability and results achieved IndFeat method: choosing only about one thousand features can provide similar classification score to the one that used all features. Similar scores were retrieved with Information Gain and Word Frequency. Definitely the worst method appears to be Odds Ratio.

Understanding dependencies between text databases, feature selection and classification algorithms is still very important challenge in modern science. We plan to use implemented feature selection methods to built feature selection committee. The committee will aggregate results provided with a different methods in order to produce set of features indicated by the majority of methods. We want to use those features in our large scale text classifier that is planned to be applied in order to improve Wikipedia category structure.

Our future work will focus on development some new approaches based on evolutionary algorithms [14] as well as some extended models of neural networks [15].

Acknowledgments. This work has been supported by the National Center for Research and Development (NCBiR) under research Grant No. SP/I/1/77065/10 SYNAT: „Establishment of the universal, open, hosting and communication, repository platform for network resources of knowledge to be used by science, education and open knowledge society”.

References

1. Li, Y., Jain, A.: Classification of text documents. The Computer Journal 41(8), 537–546 (1998)
2. Pestov, V.: On the geometry of similarity search: dimensionality curse and concentration of measure. Information Processing Letters 73(1), 47–51 (2000)
3. Blachnik, M.: Comparison of various feature selection methods in application to prototype best rules. Computer Recognition Systems 3 (2007)
4. Biesiada, J., Duch, W.: Feature selection for high-dimensional data: A kolmogorov-smirnov correlation-based filter. Computer Recognition Systems (2007)
5. Yang, Y., Pedersen, J.O.: A comparative Study on Feature Selection in Text Categorization. Morgan Kaufmann Publishers (1997)
6. Langley, P., et al.: Selection of relevant features in machine learning. Defense Technical Information Center (1994)
7. Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence 97, 273–324 (1997)
8. Forman, G.: An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research (2002)
9. Kent, J.: Information gain and a general measure of correlation. Biometrika 70, 163–173 (1983)
10. Weiss, S., Indurkhya, N.: Predictive data mining: a practical guide. Morgan Kaufmann (1998)
11. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967)
12. Szymański, J.: Wikipedia articles representation with matrix'u. In: Hota, C., Srimani, P.K. (eds.) ICDCIT 2013. LNCS, vol. 7753, pp. 500–510. Springer, Heidelberg (2013)
13. Havasi, C., Pustejovsky, J., Speer, R., Lieberman, H.: Digital intuition: Applying common sense using dimensionality reduction. IEEE Intelligent Systems 24, 24–35 (2009)
14. Balicki, J.: An adaptive quantum-based multiobjective evolutionary algorithm for efficient task assignment in distributed systems. In: Mastorakis, N.E., et al. (eds.) Recent Advances in Computer Engineering, WSEAS, Rhodes Greece, pp. 417–422 (2009)
15. Balicki, J., Z., Stateczny, Kitowski, A.: Extended hopfield model of neural networks for combinatorial multiobjective optimization problems. In: Proceedings of IEEE World Congress on Computational Intelligence - IJCNN, Anchorage, May 4-9, pp. 1646–1651 (1998)

Parallel Neuro-Tabu Search Algorithm for the Job Shop Scheduling Problem

Wojciech Bożejko[1], Mariusz Uchroński[2], and Mieczysław Wodecki[3]

[1] Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
Janiszewskiego 11-17, 50-372 Wrocław, Poland
wojciech.bozejko@pwr.wroc.pl
[2] Wrocław Centre of Networking and Supercomputing
Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
mariusz.uchronski@pwr.wroc.pl
[3] Institute of Computer Science, University of Wrocław
Joliot-Curie 15, 50-383 Wrocław, Poland
mwd@ii.uni.wroc.pl

Abstract. We propose two parallel algorithms based on neuro-tabu search method, designed to solve the jobs shop problem of scheduling. The fist algorithm is based on independent runs of the neuro-tabu with different starting points. The second one uses sophisticated diversification method based on path-relinking methodology applied to the set of elite solutions. Proposed approaches are especially effective for the instances of large size.

1 Introduction

We consider the job shop scheduling problem here, which can be described as follows (see [4]). There is a set of jobs and a set of machines. Each job consists of a number of operations which have to be processed in a given order, each one on a specified machine during a fixed time. The processing of an operation cannot be interrupted. Each machine can process at most one operation at a time. We want to find a schedule (the assignment of operations to time intervals on machines) that minimizes the *makespan*. The job shop scheduling problem, although relatively easily stated, is strongly NP-hard and it is considered as one of the hardest problems in the area of combinatorial optimization.

Because of NP-hardness of the problem heuristics and metaheuristics are recommended as 'the most reasonable' solution methods. The majority of these methods refer to the makespan minimization. We mention here a few recent studies: Jain, Rangaswamy, and Meeran [21]; Pezzella and Merelli [30]; Grabowski and Wodecki [17]; Nowicki and Smutnicki [27]; Bożejko and Uchroński [6]. Heuristics algorithms based on dispatching rules are also proposed in papers of Holthaus and Rajendran [19], Bushee and Svestka [10] for the problem under consideration. For the other regular criteria such as the total tardiness there are proposed

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 489–499, 2013.

metaheuristics based on various local search techniques: simulated annealing [34], tabu search [2] and genetic search [26].

Here we propose the new approach to the distributed tabu search metaheuristic designing to solve difficult discrete optimization problems, such as the job shop problem, using a multi-GPU cluster with distributed memory. We also determine theoretical number of processors, for which the speedup measure has a maximum value. We experimentally determine what parallel execution time T_p can be obtained in real-world installations of multi-GPU clusters (nVidia Tesla S2050 with a 6-cores CPU server) for Taillard benchmarks [32] of the job shop scheduling problem, and compare them with theoretically determined values.

2 Job Shop Problem

Job shop scheduling problems result from many real-world cases, which means that they have good practical applications as well as industrial significance. Let us consider a set of jobs $\mathcal{J} = \{1, 2, \ldots, n\}$, a set of machines $M = \{1, 2, \ldots, m\}$ and a set of operations $\mathcal{O} = \{1, 2, \ldots, o\}$. The set $\mathcal{O}$ is decomposed into subsets connected with jobs. A job j consists of a sequence of o_j operations indexed consecutively by $(l_{j-1}+1,\ l_{j-1}+2,\ \ldots,\ l_j)$, which have to be executed in this order, where $l_j = \sum_{i=1}^{j} o_i$ is the total number of operations of the first j jobs, $j = 1,\ 2,\ \ldots,\ n$, $l_0 = 0$, $\sum_{i=1}^{n} o_i = o$. An operation i has to be executed on machine $v_i \in M$ without any idleness in time $p_i > 0$, $i \in \mathcal{O}$. Each machine can execute at most one operation at a time. A feasible solution constitutes a vector of times of the operation execution beginning $S = (S_1,\ S_2,\ \ldots,\ S_o)$ such that the following constraints are fulfilled

$$S_{l_{j-1}+1} \geq 0, \quad j = 1, 2, \ldots, n, \tag{1}$$

$$S_i + p_i \leq S_{i+1}, \ \ i = l_{j-1} + 1, \ \ l_{j-1} + 2, \ldots, l_j - 1, \ \ j = 1, 2, \ldots, n, \tag{2}$$

$$S_i + p_i \leq S_j \quad \text{or} \quad S_j + p_j \leq S_i, \quad i, j \in O, \quad v_i = v_j, \quad i \neq j. \tag{3}$$

Certainly, $C_j = S_j + p_j$. An appropriate criterion function has to be added to the above constraints. The most frequent are the following two criteria: minimization of the time of finishing all the jobs and minimization of the sum of job finishing times. From the formulation of the problem we obtain $\mathcal{C}_j \equiv C_{l_j}$, $j \in \mathcal{J}$.

The first criterion, the time of finishing all the jobs

$$C_{\max}(S) = \max_{1 \leq j \leq n} C_{l_j}, \tag{4}$$

corresponds to the problem denoted as $J||C_{\max}$ in the literature. The second criterion, the sum of job finishing times

$$C(S) = \sum_{j=1}^{n} C_{l_j}, \tag{5}$$

corresponds to the problem denoted as $J||\sum C_i$ in the literature.

Both problems described are strongly NP-hard and although they are similarly modelled, the second one is found to be harder because of the lack of some specific properties (so-called block properties, see [27]) used in optimization of execution time of solution algorithms.

3 Tabu Search Mechanism with Neural Network Application

In the considered neuro-tabu search algorithm NTS each move is represented by its neuron. For the neighborhood considered in [28] a network of neurons formed of $o-1$ neurons. Let i-th neuron represents a move consisting in swap of two adjacent elements on the positions i and $i+1$ in a solution π.

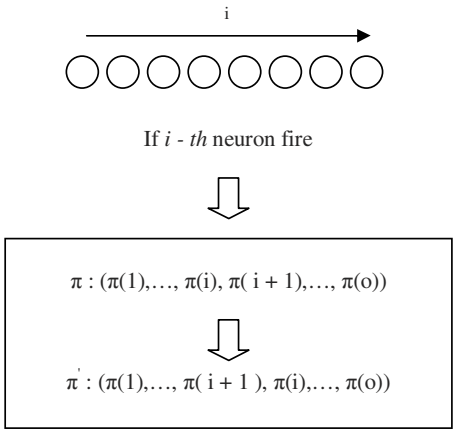


Fig. 1. Dependencies between activation of the neuron and a move

In a proposed neural network architecture a history of each neuron is stored as its internal state (*tabu effect*). If in an iteration neuron is activated, then the value 1 is fixed on its output and values 0 are fixed on the outputs of other neurons. The neuron activated in an iteration must not be activated once again for the next s iterations. Each neuron is defined by the following equations:

$$\eta_i(t+1) = \alpha \Delta_i(t), \tag{6}$$

$$\Delta_i(t) = \frac{C_{max}(\pi_v^{(t)}) - C^*_{max}}{C^*_{max}}, \tag{7}$$

$$\gamma_i(t+1) = \sum_{d=0}^{s-1} k^d x_i(t-d), \tag{8}$$

where $x_i(t)$ is an output of the neuron i in the iteration t. Symbol $C_{max}(\pi_v^{(t)})$ means the value of the goal function for the permutation obtained after executing

a move v in the iteration t, i.e. $\pi^{(t)}$. Symbol $\Delta_i(t)$ means a normalized, current value of the goal function, and C^*_{max} is the value of the best solution found so far. Parameters α i k are scale factors. A symbol $\eta_i(t+1)$ (*gain effect*) defines quality of a move v. A variable $\gamma_i(t+1)$ (*tabu effect*) stores a history of the neuron i for the last s iterations. Neuron is activated if it has a low value of the tabu effect and it gives a better reduction of the C_{max}. More detailed a neuron i is activated if it has the lowest $\{\eta_i(t+1)+\gamma_i(t+1)\}$ value of all the neurons.

If $0 < k < 1$ i $s = t$ then an equation(8) takes the form of:

$$\gamma_i(t+1) = k\gamma_i(t) + x_i(t), \tag{9}$$

where $\gamma_i(0) = 0$ and $x_i(0) = 0$ for each i. From the equation (9) it follows, that the value of $\gamma_i(t)$ of each neuron decreases exponentially (Fig. 3 and 4).

In many algorithms proposed in the literature which are based on the tabu search method a so-called aspiration criterion is implemented. It consists in executing of the forbidden move if it follows to the base solution with the goal function value lower than the best found so far.

In the proposed neuro-tabu search such a function can be implemented by ignoring tabu effect for a move v for which $\Delta_i < 0$. However during computational experiments we have observed that it does not give good effects. Therefore a proposed neuro-tabu search has not got such a function. The skeleton of the neuro-tabu search algorithm is given on the Fig. 2.

4 Parallel Neuro-Tabu Search Algorithm *pNTS*

Here we propose a solution method to the job shop problem in the distributed computing environments, such as multi-GPU clusters. Tabu search algorithm is executed in concurrent working threads, as in *multiple-walk* model of parallelization [1] (MPDS,*Multiple starting Point Different Strategies* in the Voß [33] classification of parallel tabu search metaheuristic). Additionally, MPI library is used to distribute calculations among GPUs (see Fig. 4).

Now let us consider a single cycle of the MPI data broadcasting, multi-GPU computations and batching up of the results obtained. Let us assume that the single communication procedure between two nodes of a cluster takes the time T_{comm}, the time of sequential tabu search computations is T_{seq} and the computations time of parallel tabu search is $T_{calc} = \frac{T_{seq}}{p}$ (p is the number of GPUs). Therefore, the total parallel computations time of the single cycle is

$$T_p = 2T_{comm}\log_2 p + T_{calc} = 2T_{comm}\log_2 p + \frac{T_{seq}}{p}.$$

In case of using more processors, the parallel computing time $\left(\frac{T_{seq}}{p}\right)$ decreases, whereas the time of communication $(2T_{comm}\log p)$ increases. We are looking for such a number of processors p (let us call it p^*) for which T_p is minimal. By calculating $\frac{\partial T_p}{\partial p} = 0$ we obtain

$$\frac{2T_{comm}}{p\ln 2} - \frac{T_{seq}}{p^2} = 0 \tag{10}$$

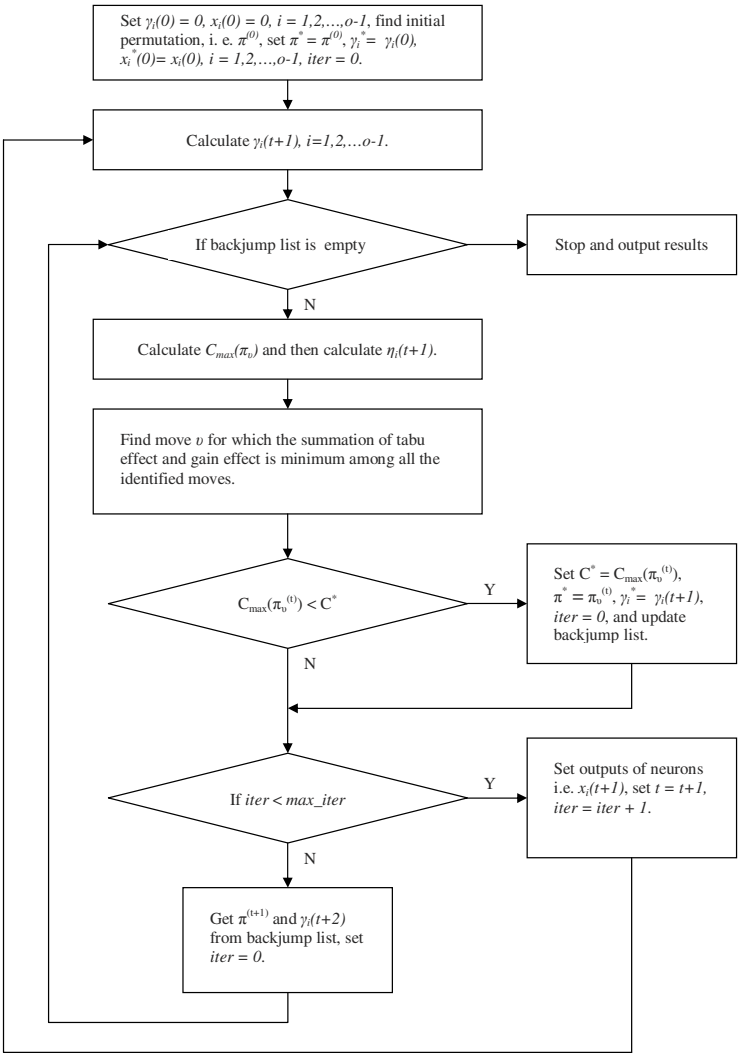


Fig. 2. A skeleton of the neuro-tabu search algorithm

and then

$$p = p^* = \frac{T_{seq} \ln 2}{2T_{comm}}, \tag{11}$$

which provides us with an optimal number of processors p^* which minimizes the value of the parallel running time T_p.

The fraction of communication time is $O(\log_2 p)$ in this tree-based data broadcasting method, therefore this is another situation than for linear-time broadcasting (discussed in [3]), for which the overall communication and calculation

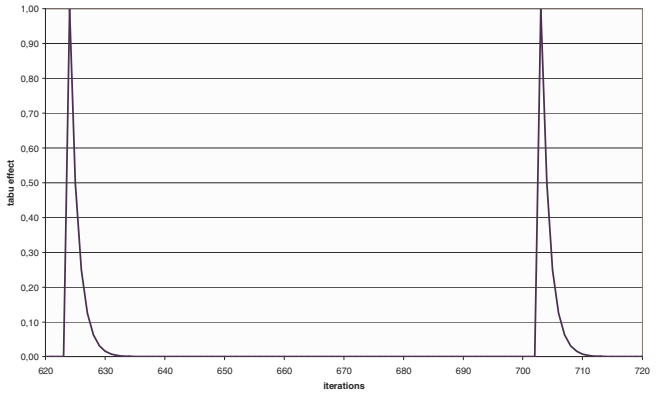


Fig. 3. Changes of the $\gamma(t)$ value for $k = 0.5$

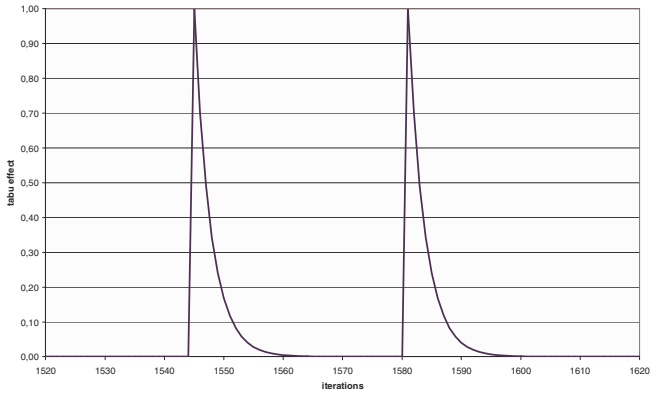


Fig. 4. Changes of the $\gamma(t)$ value for $k = 0.7$

efficiency is much lower. On the other hand the linear-time broadcasting is similar to described by Brooks' Law [9] for project management, i.e. the expected advantage from splitting development work among n programmers is $O(n)$ but the communications cost associated with coordinating and then merging their work is $O(n^2)$.

5 Advanced Neuro-Tabu Search Algorithm *iNTS*

As the second solution method we propose an approach introduced by Nowicki and Smutnicki [27] with using neuro-tabu instead of classic tabu search algorithm, as in the original paper. The proposed $iNTS$ algorithm operates on the set of dispersed (elite) solution obtained with using $NIS(\gamma, \delta, C^R)$ function (see Fig. 6), where γ, δ are two processing orders and C^R is the reference makespan (goal function value). Inside, the neighborhood generation function is used based

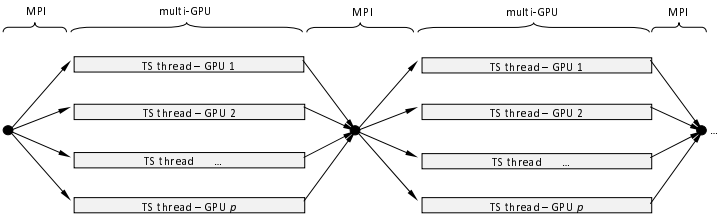


Fig. 5. Skeleton of the Multi-Level Tabu Search metaheuristic

on swap moves of adjacent operations (see Bożejko [4]). We denote by $N(\pi)$ the set of moves obtained from a solution π by swapping any two adjacent operations lying on critical path (to obtain feasible solution, see [17]). The fundamental aim of the NIS function is to provide the solution $\varphi = NIS(\gamma, \delta, C^R)$ located 'between' γ and δ reference solutions to run a neuro-tabu search explaration in the main $iNTS$ algorithm (see Fig. 7). To control the distance between any two solution α,β, the Kendall's tau measure $D(\alpha, \beta)$ is used to represent the minimal number of adjacent swap moves (inversions) to obtain permutation β^{-1} from α^{-1} (for more measures on permutations see e.g. Diaconis [13]). Values of tuning parameters were following: $maxE = 8$ (number of elite solutions), $maxD = 5$ (maximal distance used in the loop), $maxV = 0.5$.

Algorithm 1. $NIS(\gamma, \delta, C^R)$
Input: γ, δ – two processing orders; C^R – reference makespan;
Output: φ – processing order; update reference makespan C^R;
  $\pi \leftarrow \gamma$; $iter \leftarrow 0$. **Find** δ^{-1} **and** $D(\gamma, \delta)$
  repeat
    $iter \leftarrow iter + 1$; **Find** $N(\pi)$;
    For any $v \in N(\pi)$ **calculate and store** $C_{max}(\pi_{(v)})$;
    Find $N^+ = \{v = (x, y) \in N(\pi) : \delta^{-1}(y) < \delta^{-1}(x)\}$;
      if $N^+ \neq \emptyset$ **than** $K \leftarrow N^+$ **else** $K \leftarrow N(\pi)$;
      Select the move $w \in K$ **such, that**
        $C_{max}(\pi_{(w)}) = \min_{v \in K} C_{max}(\pi_{(v)})$;
      Denote $\pi_{(w)}$ **by** α;
    $\pi \leftarrow \alpha$; $\varphi \leftarrow \pi$;
    if $C_{max}(\pi) < C^R$ **than** $C^R \leftarrow C_{max}(\pi)$ **and exit**;
  until $iter \geq maxV \cdot D(\gamma, \delta)$; $\{maxV \in (0, 1)$ **- parameter**$\}$

Fig. 6. $NIS(\gamma, \delta, C^R)$ function

6 Computational Experiments

Proposed algorithms were ran on the server based on 6-cores Intel Core i7 CPU X980 (3.33GHz) processor equipped with nVidia Tesla S2050 GPU (1792 cores) working under 64-bit Linux Ubuntu 10.04.4 LTS operating system and tested on the benchmark problems taken from Taillard [32].

```
Algorithm 2. iNTS
Input: π^0 – processing orders provided by INSA;
Output: π^* – the best found processing order
              and its makespan C^*;
  Set (π^1, C^1) ← NTS(π^0) and C^* ← C^1;
  for i ← 2, ..., maxE do
      φ ← NIS(π^{i−1}, π^0, C^*); (π^i, C^i) ← NTS(φ);
      C^* = min{C^*, C^i};
  repeat
      Find 1 ≤ l ≤ maxE so that
        D(π^k, π^l) = max{D(π^k, π^i) : 1 ≤ i ≤ maxE};
        Set φ ← NIS(π^k, π^l, C^*) and (π^l, C^l) ← NTS(φ);
      if C^l < C^k than set (π^*, C^*) ← (π^l, C^l) and k ← l;
  until max{D(π^k, π^i) : 1 ≤ i ≤ maxE} < maxD.
```

Fig. 7. Algorithm $iNTS$

Computational experiments results are compared in the Table 1. Particular columns have the following notion:

- $sNTS$ – sequential Neuro Tabu Search algorithm of Bożejko and Uchroński [6],
- $pNTS$ – parallel (for $p = 16$) Neuro Tabu Search algorithm, MPSS model (due to the Voß [33] classification) without communication (proposed in the Section 4); for each processor the starting solution was generated by the NTS algorithm executed for $process\_id * 100$ iterations,
- $iNTS$ – advanced NTS algorithm based on the diversification and intensification methodology proposed in the Section 5.

Processing times for all benchmark instances were: (sequential) NTS – 132m27.319s, (parallel) $pNTS$ ($p = 4$) – 169m45.748s (longer time of parallel algorithm work causes of the method of starting solutions generation and synchronization), $iNTS$ – about 60 hours. As we can observe the proposed $iNTS$ algorithm managed to obtain the average relative percentage deviation from the

Table 1. Percentage relative deviations (PRD) to the best known solutions

problem	$n \times m$	$sNTS$	$pNTS(p = 16)$	$iNTS$
TA01-10	15×15	0.4948	0.3141	0.0652
TA11-20	20×15	1.1691	0.9129	0.4412
TA21-30	20×20	1.2486	0.7033	0.4803
TA31-40	30×15	1.0592	0.7965	0.3764
TA41-50	30×20	1.8565	1.5634	0.8328
TA51-60	50×15	0.0915	0.0915	0.0520
TA61-70	50×20	0.1479	0.0210	0.0140
TA71-80	100×20	0.0090	0.0090	0.0090
average		**0.7596**	**0.5515**	**0.2839**

best known solution of the Taillard instances on the level of 0.28%. Average PRD values for test instances of the group TA61-TA70 (1,000 operations) are 10 times lower than for the proposed $iNTS$ algorithm. Comparing results for $sNTS$ and $pNTS$ algorithm we can notice that using parallel algorithm results in obtaining much lower PRD values. Based on this we plan to implement parallel version of $iNTS$ algorithm as a future work.

7 Conclusions

In this paper we propose an approach designed to solve difficult problems of combinatorial optimization in distributed parallel architectures without shared memory, such as clusters of nodes equipped with GPU units (i.e. multi-GPU clusters). The methodology can be especially effective for large instances of hard to solve optimization problems, such as flexible scheduling problems as well as discrete routing and assignment problems.

References

1. Alba, E.: Parallel Metaheuristics. A New Class of Algorithms. Wiley & Sons Inc (2005)
2. Armentano, V.A., Scrich, C.R.: Tabu search for minimizing total tardiness in a job shop. International Journal of Production Economics 63(2), 131–140 (2000)
3. Bożejko, W.: A new class of parallel scheduling algorithms, pp. 1–280. Wroclaw University of Technology Publishing House (2010)
4. Bożejko, W.: On single-walk parallelization of the job shop problem solving algorithms. Computers & Operations Research 39, 2258–2264 (2012)
5. Bożejko, W., Uchroński: Distributed Tabu Search Algorithm for the Job Shop Problem. In: Proceedings of the 14th International Asia Pacific Conference on Computer Aided System Theory 6th to 8th Sydney, Sydney, Australia, February 6-8 (2012)
6. Bożejko, W., Uchroński, M.: A Neuro-Tabu Search Algorithm for the Job Shop Problem. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 387–394. Springer, Heidelberg (2010)
7. Bożejko, W., Uchroński, M., Wodecki, M.: Parallel Meta^{2}heuristics for the Flexible Job Shop Problem. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 395–402. Springer, Heidelberg (2010)
8. Brandimarte, P.: Routing and scheduling in a flexible job shop by tabu search. Annals of Operations Research 41, 157–183 (1993)
9. Brooks Jr., F.P.: The Mythical Man-Month, anniversary edition. Addison-Wesley, Reading (1995)
10. Bushee, D.C., Svestka, J.A.: A bi-directional scheduling approach for job shops. International Journal of Production Research 37(16), 3823–3837 (1999)
11. Crainic, T.G., Toulouse, M., Gendreau, M.: Parallel asynchronous tabu search in multicommodity locationallocation with balancing requirements. Annals of Operations Research 63, 277–299 (1995)

12. Dauzère-Pérès, S., Pauli, J.: An integrated approach for modeling and solving the general multiprocessor job shop scheduling problem using tabu search. Annals of Operations Research 70(3), 281–306 (1997)
13. Diaconis, P.: Group Representations in Probability and Statistics. Lecture Notes - Mono-graph Series, vol. 11. Institute of Mathematical Statistics, Harvard University (1988)
14. Flynn, M.J.: Very highspeed computing systems. Proceedings of the IEEE 54, 1901–1909 (1966)
15. Gao, J., Sun, L., Gen, M.: A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Computers & Operations Research 35, 2892–2907 (2008)
16. Grabowski, J.: Generalized problems of operations sequencing in the discrete production systems, Monographs 9, Scientific Papers of the Institute of Technical Cybernetics of Wrocław Technical University (1979)
17. Grabowski, J., Wodecki, M.: A very fast tabu search algorithm for the job shop problem. In: Rego, C., Alidaee, B. (eds.) Adaptive Memory and Evolution, Tabu Search and Scatter Search, Kluwer Academic Publishers, Dordrecht (2005)
18. Hanafi, S.: On the Convergence of Tabu Search. Journal of Heuristics 7, 47–58 (2000)
19. Holthaus, O., Rajendran, C.: Efficient jobshop dispatching rules: further developments. Production Planning and Control 11, 171–178 (2000)
20. Hurink, E., Jurisch, B., Thole, M.: Tabu search for the job shop scheduling problem with Multi-purpose machine, Oper. Res. Spektrum 15, 205–215 (1994)
21. Jain, A.S., Rangaswamy, B., Meeran, S.: New and stronger job-shop neighborhoods: A focus on the method of Nowicki and Smutnicki (1996). Journal of Heuristics 6(4), 457–480 (2000)
22. Jia, H.Z., Nee, A.Y.C., Fuh, J.Y.H., Zhang, Y.F.: A modified genetic algorithm for distributed scheduling problems. International Journal of Intelligent Manufacturing 14, 351–362 (2003)
23. Kacem, I., Hammadi, S., Borne, P.: Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. IEEE Transactions on Systems, Man, and Cybernetics, Part C 32(1), 1–13 (2002)
24. Pezzella, F., Morganti, G., Ciaschetti, G.: A genetic algorithm for the Flexible Job-schop Scheduling Problem. Computers & Operations Research 35, 3202–3212 (2008)
25. Mastrolilli, M., Gambardella, L.M.: Effective neighborhood functions for the flexible job shop problem. Journal of Scheduling 3(1), 3–20 (2000)
26. Mattfeld, D.C., Bierwirth, C.: An efficient genetic algorithm for job shop scheduling with tardiness objectives. European Journal of Operational Research 155(3), 616–630 (2004)
27. Nowicki, E., Smutnicki, C.: An advanced tabu search algorithm for the job shop problem. Journal of Scheduling 8(2), 145–159 (2005)
28. Nowicki, E., Smutnicki, C.: A fast taboo search algorithm for the job shop problem. Management Science 42, 797–813 (1996)
29. Pauli, J.: A hierarchical approach for the FMS schduling problem. European Journal of Operational Research 86(1), 32–42 (1995)
30. Pezzella, F., Merelli, E.: A tabu search method guided by shifting bottleneck for the job-shop scheduling problem. European Journal of Operational Research 120, 297–310 (2000)

31. Pinedo, M.: Scheduling: theory, algorithms and systems. Prentice-Hall, Englewood Cliffs (2002)
32. Taillard, E.: Benchmarks for basic scheduling problems. European Journal of Operational Research 64, 278–285 (1993)
33. Voß, S.: Tabu search: Applications and prospects. In: Du, D.Z., Pardalos, P.M. (eds.) Network Optimization Problems, pp. 333–353. World Scientific Publishing Co., Singapore (1993)
34. Wang, T.Y., Wu, K.B.: An eficient configuration generation mechanism to solve job shop scheduling problems by the simulated annealing. International Journal of Systems Science 30(5), 527–532 (1999)

Matrix Factorization for Travel Time Estimation in Large Traffic Networks

Krzysztof Dembczyński[1], Wojciech Kotłowski[1], Przemysław Gaweł[1,2], Adam Szarecki[2], and Andrzej Jaszkiewicz[1]

[1] Institute of Computing Science, Poznań Univesity of Technology, Piotrowo 2, 60-965 Poznań, Poland
[2] NaviExpert sp. z o.o., Dobrzyckiego 4, 61-692 Poznań, Poland

Abstract. Matrix factorization techniques have become extremely popular in the recommender systems. We show that this kind of methods can also be applied in the domain of travel time estimation from historical data. We consider a large matrix of travel times in which the rows correspond to short road segments and the columns to 15 minute time slots of a week. Then, by applying matrix factorization technique we obtain a sparse model of latent features in the form of two matrices which product gives a low-rank approximation of the original matrix. Such a model is characterized by several desired properties. We only need to store the two low-rank matrices instead of the entire matrix. The estimation of the travel time for a given segment and time slot is fast as it only demands multiplication of the corresponding row and column of the low-rank matrices. Moreover, the latent features discovered by the matrix factorization may give an interesting insight to the analyzed problem. In this paper, we introduce that kind of the model and design a fast learning algorithm based on alternating least squares. We test this model empirically on a large real-life data set and show its advantage over several standard models for travel estimation.

1 Introduction

Travel time estimation is a fundamental and important part of many traffic-related systems. For example, the online personal car navigation requires the travel time estimates for all road segments in the traffic network for finding the shortest, or better to say, the fastest path between any two points in the network. Therefore, to get reasonable travel time estimates, we need to use a source of data that is able to cover the entire traffic network, or at least a part of it. The common approaches to travel time estimation are still based on loop detectors [13] or some other stationary sensors [7], so the majority of the current research focuses on single paths [13], freeways [12], or a subset of urban arterial roads [2]. Thus, we have to rely on other sources of data such as GPS-devices installed in the vehicles. The usage of GPS data in travel time prediction [4] is still a novel approach. The GPS data have an advantage of covering a large part of the traffic network, however, they are unfortunately sparse and unevenly distributed, what makes the prediction and estimation much harder.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 500–510, 2013.

Let us assume that our goal is to perform accurate estimates for each single road segments and 15 minute time slots of a week. Different models can be used for this task. The simplest method may rely on averaging the travel times for a given road segment and a time slot. However, such an approach would require a lot of observations to deliver reliable estimates and to store estimates for each segment and time slot. A remedy for these problems is to use some more general features of road segments and time slots or to compute estimates on less grain level, for example, for a given road segment. There are also many other statistical and learning methods that have already been considered for travel time estimation from GPS data, like Kalman filters [8], ARIMA models [2], linear regression [10], neural networks [13] and copula based estimation [14], to mention just a few of them. In this paper, however, we follow another way.

We show that matrix factorization techniques that have become extremely popular in recommender systems [11,9] can be successfully applied to the task of the travel time estimation from historical data in a large traffic network. Typically, in the recommender system applications, the rows of the matrix correspond to users, while the columns to products, movies, or songs. Such a matrix is sparse, since it is rather unlikely that a given user would buy all possible products or watch all the movies. This matrix is then approximated by a product of two low-rank matrices that represent latent features of users and products. In travel time estimation, the rows of the matrix may correspond to short road segments and the columns to 15 minute time slots of a week. Then, by applying matrix factorization techniques we obtain a sparse model in which the latent features describe road segments and time slots. Such a model is characterized by several desired properties. We only need to store the two low-rank matrices instead of the entire matrix. The estimation of the travel time for a given segment and time slot is fast as it only demands multiplication of the corresponding row and column of the low-rank matrices. Moreover, the latent features discovered by the matrix factorization may give an interesting insight to the analyzed problem. By using a specific regularization over the time slots we can obtain smooth features that represents time characteristics of segments.

In this paper, we introduce that kind of the model and design a fast learning algorithm based on alternating least squares. We test this model empirically on a large real-life data set and show its advantage over several standard models for travel estimation.

2 Problem Statement

The goal is to predict a travel time y_{st} for a given road segment $s \in \{1, \ldots, S\}$ in a given time point t. The task is then to find a function $f(s,t)$ that predicts accurately the value of y_{st}. The accuracy of a single prediction $\hat{y}_{st} = f(s,t)$ is measured by a loss function $L(y_{st}, \hat{y}_{st})$ which determines the penalty for predicting $\hat{y}_{st}$ when the true value is y_{st}. We will use the squared error as the loss function: $L(y_{st}, \hat{y}_{st}) = (y_{st} - \hat{y}_{st})^2$. The set of historical data $\{(y_i, s_i, t_i)\}_{i=1}^N$ is used by a learning procedure to construct function $f(s,t)$ in order to minimise

loss over future data. We additionally assume that for each road segment s and time point t, a vector of attributes is available: $\mathbf{x}_{st} = (x_{st1}, x_{st2}, \ldots, x_{stn})$. Thus the data can be represented in regular tabular form $\{(y_i, \mathbf{x}_i)\}_{i=1}^{N}$. Exemplary attributes include: segment id, type of the road, type of surrounding area, geographical information and segment length (which will be denoted as l_s).

Below, we first discuss simple granular models and Bayesian averaging of them as two baseline models. Then, we introduce the variant of matrix factorization suited for travel time estimation.

3 Granular Models

The granular model is based on averaging over specified granulation of data. A granule can be defined by a conjunctive rule G_m:

$$G_m(\mathbf{x}) = \prod_{j=1}^{n} [\![x_j \in S_j]\!]$$

where S_j is a subset of a domain of j-th attribute, and $[\![P]\!]$ is 1 if predicate P is satisfied, 0 otherwise. In other words, $G_m(\mathbf{x})$ indicates whether $\mathbf{x}$ belongs to the granule being the intersection of conditions $x_j \in S_j$. All granules $\{G_m\}_1^M$ are disjoint and cover the entire feature space, i.e., for any $\mathbf{x}$ there exists only and exactly one m for which $G_m(\mathbf{x}) = 1$. The prediction is computed as:

$$f(\mathbf{x}_{st}) = l_s \sum_{m=1}^{M} \alpha_m G_m(\mathbf{x}_{st}), \quad \alpha_m = \frac{\sum_{i=1}^{N} y_i G_m(\mathbf{x}_i)}{\sum_{i=1}^{N} l_i G_m(\mathbf{x}_i)} \tag{1}$$

where M is number of granules and α_m is an estimate of a travel time for a length unit computed as an average over training observations belonging to the m-th granule. The main problem is to determine the right granulation, to achieve the desired bias-variance trade-off, with coarse granules yielding biased predictions, while fine granules may yield high variance. In this work we use several simple groupings of attribute values for granules.

One can notice that formula (1) resembles prediction function used in decision tree [3] and rule models [6,5]. The main difference is that functions G_m are not directly induced, but given a priori and based on simple grouping of attribute values. Since in the considered case, there are only few attributes available and some of them are of specific kind like road segment id (a nominal attribute with a huge number of values) such an approach seems to be reasonable.

4 Bayesian Averaging

Bayesian averaging allows combining two models built on different levels of granulation, based on the variance and the number of observations in the granules. Let f_{s1} and f_{s2} be two granular models. The first model f_{s1} is assumed to be

computed on coarse granules and can be treated as a prior expectation for a predicted value. In turn, f_{s2} gives a fine-grain prediction. Bayesian averaging is then defined by

$$f_s(\mathbf{x}) = (1-\lambda)f_{s1}(\mathbf{x}) + \lambda f_{s2}(\mathbf{x}),$$

where λ might be tuned empirically. However, we can determine λ using the Bayesian analysis as follows. Let μ_0 be the mean travel time within the coarse granule m_1. We assume that the mean travel time μ for each fine granule m_2 within coarse granule m_1 is drawn from a Gaussian distribution with mean μ_0. Then, the observations y_i in the fine granule m_2 are drawn from a Gaussian distribution with mean value μ. If the parameters of the distributions are estimated from the data, one can show [1], that the Bayesian posterior would lead to the following value of λ:

$$\lambda = \frac{\hat{\sigma}^2_{m_1}}{\hat{\sigma}^2_{m_1} + \hat{\sigma}^2_{m_2}/n_{m_2}}, \tag{2}$$

where $\hat{\sigma}^2_{m_1}$ is the variance of all observations within the coarse granule G_{m_1}, while $\hat{\sigma}^2_{m_2}$ is the variance of observations within the fine granule G_{m_2}, and n_{m_2} is number of observations in granule G_{m_2}. If the term $\hat{\sigma}^2_{m_2}/n_{m_2}$ is small comparing to $\hat{\sigma}^2_{m_1}$ (many observations and/or large variance within the coarse granule), then f_{s2} tends to dominate the final prediction, whereas if the term is high, then the overall predictions is strongly shrunk towards the coarse-grained model f_{s1}.

5 Matrix Factorization

Let us consider a matrix Y, which rows correspond to road segments and columns to 15 minute time slots defined over a week (672 time slots in total). The task is to find a compressed (storage efficient) an approximation of Y that also has all of the missing values from Y supplemented (which may be many due to sparsity of the data).

The approximation of Y can be given by a product of two low-rank matrices: $Y \simeq \hat{Y} = UM^T$ where U is an $S \times K$ and M is a $J \times K$ matrix, where S is the number of road segments, and J the number of time slots. K defines a rank of $\hat{Y}$ and can be seen as a number of latent features describing road segments and time slots.

The elements y_{sj} of matrix Y are mapped from training observations $\{(y_i, s_i, t_i)\}_{i=1}^N$, in such a way that a time point t_i is transformed to a corresponding time slot j. We will use $i \mapsto sj$ to express this mapping. Let us also note that elements of Y may, in fact, contain more than one entry, and many of the elements will contain no entry at all.

Formally, we try to find optimal matrices $U^* = [u_{sk}]$ and $M^* = [m_{jk}]$, which are the solution of:

$$(U^*, M^*) = \arg\min_{(U,M)} \mathcal{L}(Y, UM^T),$$

where

$$\mathcal{L}(Y, UM^T) = \sum_{i \mapsto sj} (y_{sj} - \sum_{k=1}^{K} u_{sk} m_{jk})^2.$$

In the above formulation, the rank K of matrix $\hat{Y}$ plays a role of regulariser that prevents overfitting of the model to training data. An additional regularisation is usually introduced in this type of algorithms. For example, the Frobenius norm of matrices U and M:

$$\sum_{sk} u_{sk}^2, \qquad \sum_{jk} m_{jk}^2,$$

respectively, is usually minimised along with the error term. This prevents the coefficients to have values far from zero.

The Frobenius regularisation for segments, however, requires some calibration of the matrix entries to justify regularization to zero. In the following, we set up the first vector of U to be equal to the original length of the segment: $u_{s1} = l_s$. Values for m_{j1} can be then interpreted as average travel times for a length unit (i.e., average inverse velocity).

In turn, for time slots we control the difference between consecutive time slots, as the time slots close to each other should result in a similar prediction. This can be achieved by smoothing a prediction over time slots using a specific regularisation term of the following form:

$$\sum_{jk} (m_{jk} - m_{(j-1)k})^2 = \sum_{jk} m_{jk}^2 + m_{(j-1)k}^2 - 2 m_{jk} m_{(j-1)k},$$

where we assume that if $j = 1$ then $j - 1 = J$, as also if $j = J$, then $j + 1 = 1$. Thus, the function $\mathcal{L}$ to be minimised is given by:

$$\mathcal{L}(Y, UM^T) = \sum_{i \mapsto sj} (y_{sj} - \sum_{k=1}^{K} u_{sk} m_{jk})^2 + \lambda_1 \sum_{sk} u_{sk}^2 + \lambda_2 \sum_{sk} (m_{jk} - m_{(j-1)k})^2$$

The parameters of the problem are then K, as also λ_1 and λ_2 that controls the strength of regularisation.

The regularised learning problem is unfortunately non-convex, thus the minimisation of our objective $\mathcal{L}(Y, UM^T)$ requires a special method to solve. Our method is iterative. We first set up $u_{s1} = l_s$ for all s. Then, in each iteration $(k = 1, \ldots, K)$ we compute u_{sk} (except u_{s1}) and m_{jk}, for all s and j, by minimising:

$$\mathcal{L}^k = \sum_{i \mapsto sj} (\Delta y_{sj} - u_{sk} m_{kj})^2 + \lambda_1 \sum_{s} u_{sk}^2 + \lambda_2 \sum_{j} (m_{jk} - m_{(j-1)k})^2,$$

where

$$\Delta y_{sj} = y_{sj} - \sum_{k'=1}^{k-1} u_{sk'} m_{jk'}.$$

Minimising $\mathcal{L}^k$ is still hard and requires an iterative procedure, which is guaranteed to converge to a local minimum. In each iteration of the procedure, we first find the optimal u_{sk}, for all s, given fixed m_{jk}. Next, we fix new values of u_{sk} and find the optimal m_{jk} for all j. This process is repeated until convergence, and usually only several iterations are sufficient to get a stable solution.

The solution for u_{sk} given fixed m_{jk} can be obtained by taking the negative gradient of $\mathcal{L}^k$ and setting it to 0:

$$-\frac{\partial \mathcal{L}}{\partial u_{sk}} = 2\sum_j (\Delta y_{sj} - u_{sk} m_{jk}) m_{jk} - 2\lambda_1 u_{sk} = 0 \quad s = 1, \ldots, S.$$

It is easy to see that the solution is given by:

$$u_{sk} = \frac{\sum_j \Delta y_{sj} m_{jk}}{\sum_j m_{jk}^2 + \lambda_1} \quad s = 1, \ldots, S. \tag{3}$$

The penalty term for m_{jk} is more complex and couples consecutive time slots. The negative gradient is given by:

$$-\frac{\partial \mathcal{L}}{\partial m_{sj}} = 2\sum_s (\Delta y_{sj} - u_{sk} m_{jk}) u_{sk} - 2\lambda_2 (2m_{jk} + m_{(j-1)k} + m_{(j+1)k}),$$

where $j = 1, \ldots, J$. By setting all negative gradients to 0 we end up with the following system of linear equations:

$$m_{jk} + \frac{\lambda_2 (m_{(j-1)k} + m_{(j+1)k})}{2(\sum_s u_{sk}^2 + \lambda_2)} = \frac{\sum_s \Delta y_{sj} u_{sk}}{\sum_s u_{sk}^2 + \lambda_2} \quad j = 1, \ldots, J. \tag{4}$$

We solve this system of linear equation by using Gauss-Seidel method which proceeds in an iterative way.

6 Experimental Results

6.1 Data and Methodology

Real GPS floating car data used in the experiment were delivered by NaviExpert, a Polish car navigation company. The data consist of map-matched (projected to road segments) travel time observations, associated with a time stamp and additional attributes such as the length of a road segment, road category (highway, freeway, urban road, etc.), type of surrounding area (city, village, out-of-the-city), geographical coefficients, etc.

The observations cover the city of Warsaw, the capital of Poland, with surroundings — a rectangular envelope with a side of about 85km around the centre at 52.2391°N 21.0227°E, which constitutes an area of above 7000km^2. The observations span from the 1st of September 2009 to the 31st of December 2009. In total, the data set contains 6 808 061 observations that are sparse and unevenly distributed in time and space.

Table 1. RMSE for static models with varying size of training set: we used 6.25%, 12.5%, 25%, 50%, 100% of training instances. RMSE is given in percent ([%]) in comparison to the global mean model and in minutes ([min])

model	6.25%N		12.5%N		25%N		50%N		100%N	
	[%]	[min]	[%]	[min]	[%]	[min]	[%]	[min]	[%]	[min]
GM	100.00	0.391	100.00	0.390	100.00	0.391	100.00	0.394	100.00	0.395
LLG	95.94	0.375	96.08	0.374	95.90	0.375	95.55	0.376	95.37	0.377
RSG	89.97	0.352	88.57	0.345	87.01	0.340	85.27	0.336	83.70	0.331
BA	89.42	0.349	88.45	0.345	87.06	0.340	85.43	0.336	84.00	0.332
MF	89.69	0.351	87.67	0.342	84.84	0.332	82.37	0.324	80.42	0.318

The models were trained and tuned on the training set. Then, the resulting models were evaluated on the testing set. The training set covered the first three months of data while the test set covered the whole month of December 2009. As performance measure we used the root mean square error (RMSE).

6.2 Results

In the experiment we compared the following four models: low-level granular (LLG) model, single road segment (SRG) model, Bayesian averaging (BA) of two above models, and the matrix factorization (MF). All the results are related to the simple global mean (GM) model, for which the average of travel times was computed over all training observations.

LLG is a simple granular model in which a granule is built as a combination of values of the following attributes: time periods (possible values: morning, noon, afternoon, nights-and-weekends), type of the road (possible values: highway, main road, normal), and type of surrounding area (possible values: town, village, out-of-city). SRG is also a simple granular model which prediction is computed as average travel time over all observations from a given road segment. In case of a low traffic on a segment, the prediction may have a high variance, which is the main drawback of this model. To overcome the above problem, we used BA to combine LLG and SRG models with each other.

Obviously, MF is the most complex model used in this experiment that requires tuning of additional parameters. In this regards, we isolated from the training set a validation part that contained observations from November. However, computational costs of this algorithm are not very high. The single run of the algorithm takes around 20 seconds on Inter Core2 2.4GHz machine with 2 GB of RAM. We computed up to $K = 20$ factors for matrices U and M. We found the optimal value $K = 10$ on the validation set. Regularisation parameters were also optimized on the validation set.

Table 1 shows the results given in plain RMSE (in minutes) and the percentage RMSE in relation to the GM model's prediction error for different sizes of

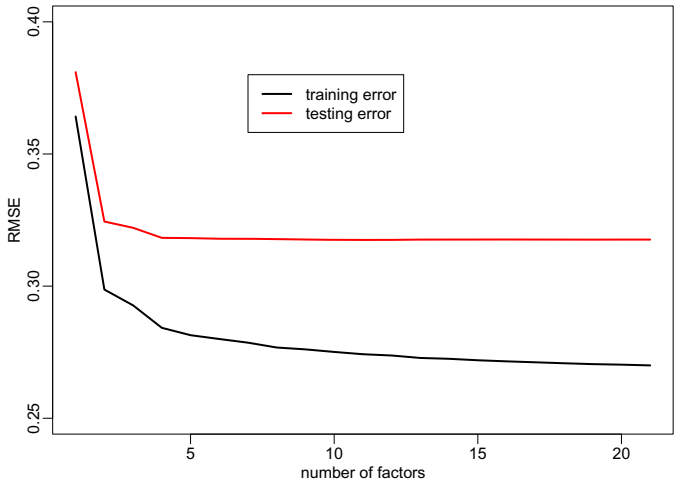


Fig. 1. Training and testing RMSE of MF model are given as a function of K, i.e., number of factors of matrices U and M

training set. We varied the size of training set by taking from 6.25% to 100% of training instances. One can easily observe that the MF model performs the best. In the case of small training set, BA has a small advantage over MF and SRG. If the size of training size increases, then SRG and MF starts to outperform BA, however, in the case of the latter the improvement is more pronounced. These three models are significantly better than GM and LLG models. In Figure 1 we show performance on training and testing set of MF with respect to number of factors K. Significant improvement can be noticed up to 5 factors. For next factors, RMSE computed on testing set does not decrease, however, no overfitting occurs.

Let us also underline that the MF model has an additional advantage that the latent features discovered by matrix factorization can be nicely interpreted. Figure 2 shows the first 5 factors of matrix M and distribution of values on the corresponding 5 factors of matrix U. The first factor (black) can be interpreted as a average travel time for a length unit (inverse average velocity). The next factors can be interpreted as changes in travel times depending on a type of a road segment. The second factor (red) seems to indicate road segments that are sensitive for traffic congestion in morning and afternoon hours of working days. The third (green) and fourth (blue) factor indicate in- and out-of-city segments, while the fifth factor (light-blue) indicates "Friday afternoon and weekend" roads.

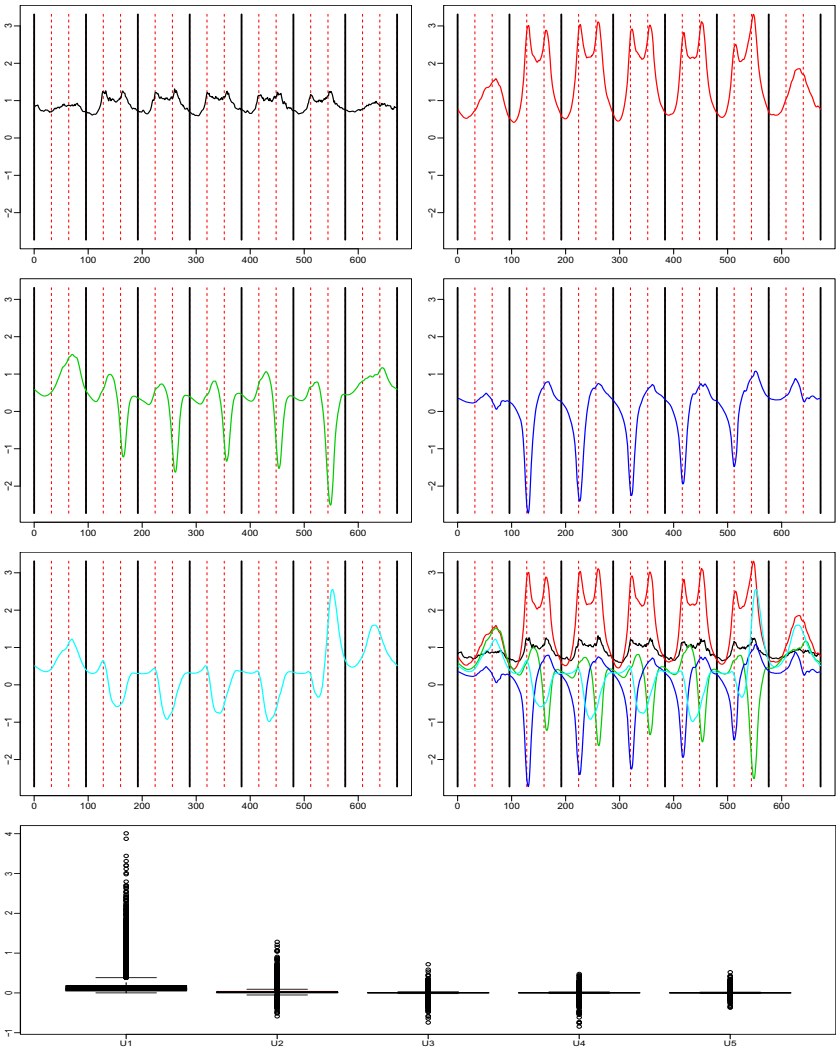


Fig. 2. From top: first five factors of matrix M that represents different characteristics of road segments. A unit of x-axis corresponds to an interval of 15 minutes (thus, a range of the x-axis is from 1 to 672). At bottom: box plot (with outliers) that shows distribution of values on corresponding factors of matrix U.

7 Conclusions

We shown that matrix factorization techniques can be successfully applied in the context of travel time estimation for large traffic networks. In the exhaustive experiment on real-world data the matrix factorization outperformed standard approaches to this problem being still competitive in terms of computational costs. Moreover, the latent features obtained as a by-product of the matrix factorization give us further insight into the problem and can be valuable source of information. There are also many possible directions for the future research. The static model could be improved by treating some road segments separately. Specific segments with dense data could also be modelled differently, e.g. by kernel estimation methods or locally weighted regression.

Acknowledgements. This work has been supported by the Polish National Science Centre, grant no. N N519 441939.

References

1. Berger, J.O.: Statistical Decision Theory and Bayesian Analysis, 2nd edn. Springer (1993)
2. Billings, D., Yang, J.: Application of the ARIMA Models to Urban Roadway Travel Time Prediction-A Case Study. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2006, vol. 3 (2006)
3. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth (1984)
4. Brosch, P.: A service oriented approach to traffic dependent navigation systems. In: IEEE Congress on Services - Part I, pp. 269–272 (2008)
5. Dembczyński, K., Kotłowski, W., Słowiński, R.: Ender - a statistical framework for boosting decision rules. Data Mining and Knowledge Discovery 21(1), 52–90 (2010)
6. Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles. Annals of Applied Statistics 2(3), 916–954 (2008)
7. Hiramatsu, A., Nose, K., Tenmoku, K., Morita, T.: Prediction of travel time in urban district based on state equation. Electronics and Communications in Japan 92(7), 1–11 (2009)
8. Liu, H., Lint, H.v., Zuylen, H.v., Zhang, K.: Two distinct ways of using kalman filters to predict urban arterial travel time. In: IEEE Intelligent Transportation Systems Conference, ITSC 2006., pp. 845–850. IEEE (2006)
9. Paterek, A.: Improving regularized singular value decomposition for collaborative filtering. In: Proc. KDD Cup Workshop at SIGKDD 2007, 13th ACM Int. Conf. on Knowledge Discovery and Data Mining, pp. 39–42 (2007)
10. Rice, J., Zwet, E.V.: A simple and effective method for predicting travel times on freeways. IEEE Transactions on Intelligent Transportation Systems, 5(3), 200–207 (2004)
11. Srebro, N., Rennie, J., Jaakkola, T.: Maximum Margin Matrix Factorizations. In: Proc. of Advances in Neural Information Processing Systems (2005)

12. Van Lint, J.: Reliable real-time framework for short-term freeway travel time prediction. Journal of Transportation Engineering 132, 921 (2006)
13. Van Lint, J., Hoogendoorn, S., Van Zuylen, H.: Accurate freeway travel time prediction with state-space neural networks under missing data. Transportation Research Part C 13(5-6), 347–369 (2005)
14. Wan, K., Kornhauser, A.: Turn-by-turn routing decision based on copula travel time estimation with observable floating-car data. In: Transportation Research Board 89th Annual Meeting. No. 10-2723 (2010)

Semantically-Driven Rule Interoperability – Concept Proposal*

Krzysztof Kaczor and Grzegorz J. Nalepa

AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
{kk,gjn}@agh.edu.pl

Abstract. Despite the maturity of the rule-based technologies, rule interoperability still constitutes an existing problem. Currently, there are many different rule representations and languages. Some of them provide an underlying formalism, while the others are just programming solutions. The variety of the representations make the interoperability process difficult. This paper is a part of research aiming at definition of formalized rule representation model, which is intended to capture semantics of the rule-based knowledge in most common representations.

1 Introduction

Rule-Based Systems (RBS) [1,2] constitute a mature technology in the field of Artificial Intelligence. Over the years, they were applied as the decision support systems in many domains like medicine, engineering, etc. Nowadays, classic RBS do not directly fit into directions and trends of the current scientific research. Nevertheless, they constitute a source of ideas and solutions for new technologies like Business Rules (BRs), Semantic Web, etc. Despite the maturity of the RBS, there are still unsolved problems and open research issues which must be taken into account in the context of the new technologies [3].

Our work focuses on one of these problems, which is related to Business Rules (BRs) [2,1] interoperability. The rule interoperability problem involves rules translation from one representation to another with the preservation of their semantics. This is a very important problem especially nowadays when the number of rule-based technologies and tools is constantly growing. The lack of an efficient rule interoperability methods makes the knowledge interchange between existing tools and technologies impossible. This problem seems to be easy to solve, but going deeper, it must take many issues into account: knowledge semantics, structure of a rule base, syntax of a rule language and inference issues. On the other side, currently, there are many different rule representations. Some of them are just programming solutions providing only rule language a without precisely defined semantics. In turn, other provide an underlying formalism, which is usually based on a logic that defines knowledge semantics. Considering

* The paper is supported by the SaMURaI Project funded from NCN (National Science Centre) resources for science DEC-2011/03/N/ST6/00886.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 511–522, 2013.

the variety of rule languages and existing logics: first order logic, propositional logic, different temporal, model and deontic logics, the translation between any two representations becomes difficult. Existing tools and technologies try to provide support for common rule representations. Such approaches results in a very general and complex technologies that still face many problems (see Section 3).

Our research aims at providing a framework for efficient and formalized rule interoperability method that is supported by tools. As a part of our research, we are going to define a formalized rule representation model which takes the most important aspects of the existing rule representations into account. Thanks to the model, we will be able to provide a formal definition of the rule languages that are only a programming solutions as well as provide a unified definition of the knowledge semantics for different rule representations. This will later allow for definition of the formalized rule interoperability methods. As a starting point, the formalization of the XTT2 (eXtended Tabular Trees version 2) rule language [4] is considered, because it provides a rigorous formal model that is based on the ALSV(FD) (Attributive Logic with Set of Values over Finite Domains) logic [1]. On the other hand, the expressiveness of the XTT2 rule language is too weak to support many rule representations. This is why, before the definition of the model, an analysis of the existing rule representations must be done in order to identify the most important features that must be included within the model.

This is a work in progress paper that tries to identify the required features of the model. The features are identified by analysis of the Drools Rule Language (DRL) from two different perspectives: dynamic and static. First of them is related to logical interpretation of a knowledge base. Through the second one, we refer to rule language syntax that is used for specification of a rule base. This analysis is an original contribution of this paper. In [5] other approach to rule base analysis is presented. It takes knowledge semantics, structure of a knowledge base and rule language syntax into account as three different levels of a rule interoperability. This paper provides a fusion of the results stemming from analysis of the CLIPS and Drools rule languages.

This paper is organized as follows: In Section 2 a motivation for our research is discussed. What is more, this section introduces possible application areas for efficient interoperability methods. The Section 3 provides an overview of the current rule interoperability technologies and approaches. The short conclusion and future work are presented in Section 5.

2 Motivation

An efficient interoperability method can find an application in many contexts. The following part of this section provides description of several important areas where an efficient interoperability method can be applied.

The first and the most obvious is the *knowledge translation* from one representation to another. Having an efficient interoperability method, one can translate rules into another representations and use within other contexts, domains, etc. One can also use different tools for rules processing and inference, what allow for

using knowledge in a different way. What is more, this also makes the knowledge sharing and deployment easier.

The another application area involves *collective knowledge development*. Thanks to interoperability methods taking rule semantics into account, it is possible to create a knowledge repository, where knowledge is stored in different representations. Every knowledge engineer can design rules by using most convenient technology [6]. Later, the knowledge is processed by an engine that translates all stored rules into one representation and performs inference. This application area is important nowadays, because the collaboration between knowledge engineers becomes more and more common.

Third application area for formalized interoperability methods is related to *knowledge meaning comparison*. Knowledge, encoded in different representations, can be compared by using a unified formal model. Knowledge comparison is a complex task, because it must be formally proven that the knowledge before and after translation has the same semantics. This is why, an interoperability method must provide a formalized underlying model. What is more, having a formalized and unified knowledge representation model, such comparison of the expressiveness of rule representation languages becomes possible. However, this issue requires a separate thread of research.

The *knowledge base verification* is the another application area for the rule interoperability methods. Nowadays, the quality of the rule base is very important. The logical errors in the rule base can lead to system failures, what in turn can expose people on different losses e.g. the financial or health. The rule base verification tools, that are dedicated for some representation, can be used for verification of the knowledge translated from other representations. This can be done transparently for the knowledge engineer or user by using automated process of translation, verifications and reporting.

3 State of the Art

Considering rule interoperability problem, it is important to provide an overview of the existing technologies that were invented to solve it. This section briefly introduces the most common technologies for knowledge interoperability.

Business Rules are one of the latest application field for the classic rules. They are intended to be used by non engineers people for defining logical aspects of business. BRs semantics is defined by Semantics of Business Vocabulary and Business Rules (SBVR) standard [7]. The goal of SBVR is to provide the basis for formal natural language declarative specifications of business entities (vocabulary) and policies (rules). The formal representation is based on several logics including first order logic, alethic modal logic and deontic logic. Furthermore, it adapts model theoretic interpretations for semantic formulations.

SBVR provides a complex meta-model, consisting of four submodels which define the representation of the business and business rules vocabulary. Due to its complexity, SBVR is hard to use in practical applications. What is more, the support for SBVR requires a dedicated parsers of controlled English.

The Semantic Web initiative aims at adding meaning to the data in order to make it machine processable. This allows for using more advanced searching and planning mechanisms, which work in an automated way. Hence, the Semantic Web mainly focuses on ontology modeling. Ontologies are designed using the OWL language [8] which provides an XML-based syntax and semantics described by Description Logics (DL) [9].

One of the main problems in Semantic Web is efficient reasoning and application of rules. One of the approach to use rules in Semantic Web is the Semantic Web Rule Language (SWRL) [10]. It combines the OWL sublanguages (OWL DL and OWL Lite) with the RuleML [11] (Datalog). It includes an abstract syntax for Horn rules in the mentioned sublanguages of OWL. Nevertheless, tool support for SWRL is still limited to DL-Safe subset of SWRL. The new OWL 2 RL Profile is a rule subset of OWL 2 that allows for using simple rule semantics with OWL 2. However, the current support for this feature in DL reasoners as well as Semantic Web using them is very limited [12].

The Semantic Knowledge Engineering (SKE) [13] approach provides a formalized model for classic rules representation. It is based on the ALSV(FD) logic [1]. The SKE design methodology is supported by a set of dedicated tools allowing for visual rules modeling, quality analysis and automated implementation [13].The underlying logic provides support for complex types and generalized attributes.This makes the logic more expressive than other rule base representations. The SKE methodology provides dedicated and visual logic-based rule language called Extended Tabular Trees version 2 (XTT2) [4].

The XTT2 is based on the ALSV(FD) logic, however it does not use it in an efficient way. It provides several limitations: it does not support complex types, it allows for using only constant expressions within conditional part of rules. Moreover, SKE assumes the underlying ALSV(FD) logic is dedicated for the XTT2 rule language. It does not consider if the provided formalism can be used for rule interchange purposes.

Apart from the nowadays approaches to knowledge base design, there were several attempts done in the past. Historically, an important approach that aimed at full formalization of the knowledge representation including basic inference tasks was KADS. In fact, in this area an important effort was done, see [14,15]. However, KADS had a very broad perspective on the knowledge representation, with rules being one of several methods. Due to this complexity, KADS-oriented research did not result in any practical tools for RBS.

Currently several solutions are being proposed for knowledge interchange: Rule Interchange Format (RIF), Rule Markup Language (RuleML), REVERSE Rule Markup Language (R2ML), and Knowledge Interchange Format (KIF).

RIF [16] is indented to be an extensible interchange format for all rule languages, mainly for Semantic Web. The architecture of RIF consists of several dialects, which are the XML-based rule languages with well-defined semantics. Some of the dialects, especially RIF Core, are at early stage of development and can be superseded. The practical application of RIF is limited because of their complexity and generic nature. What is more, the RIF specification leaves

many questions open e.g. how to implement a transformation from a source rule language into the RIF. Currently, a number of tools tries to provide support for RIF e.g. RIFLE[1], SILK[2], fuxi[3], and many others. Nevertheless, none of these tools provide full support for RIF.

RuleML [17] is an XML-based language for rules representation. It provides a way of expressing BRs in modular stand-alone units and allows for the deployment, execution, and exchange of rules between different systems and tools. Each unit provides support and semantics for a specific rule language and application. Thanks to the modular units, RuleML seems to be very flexible and extensible. However, RuleML does not provide any mechanisms for semantics evaluation of the rule language elements. This is why, the rule interchange can be inconsistent at the semantic level. Furthermore, the provided units for specific rule languages lead to emergence of dialects. Thus, RuleML suffers from the same problem as RIF. This is why, an implementation of tools that support all the RuleML dialects becomes practically impossible. Moreover, the issue of interchange between rule languages with complex data types and vocabularies (Drools) is not well clarified in RuleML.

R2ML [18] is a visual rule markup aiming at capturing formalization of different rule languages in order to enable rule interchange. It also allows for enriching ontologies by rules. It has a XML based concrete syntax validated by an XML Schema allowing for different semantics for rules. Rule concepts are defined with the help of MOF/UML — a subset of UML class modeling language. Later, the MOF/UML representation is mapped to the given markup syntax. In general, R2ML does not provide any predefined semantics. This is why, semantics has to be defined every time for each rule language. What is more, R2ML does not ensure that a defined interchange method is lossless.

KIF[4] is a computer-oriented language for the interchange of knowledge among different applications. It is logically comprehensive (i.e. it provides the expression of arbitrary sentences in the first-order predicate calculus) and provides declarative semantics. It is possible to understand the meaning of the expressions without usage of interpreter. In this way, KIF differs from other languages that are based on specific interpreters such as Emycin and Prolog. The KIF language was intended as highly specialized not efficient knowledge representation language. So, it should not be used as an internal knowledge representation for applications. KIF and their successors (SUO-KIF, LKIF) suffer from the same reasons as SBVR. The specification of KIF is very complex and vague. Similarly to SBVR, KIF provides very complex meta-model consisting of large number of classes. Hence, from the practical point of view, complexity of the meta-model causes that there is a lack of tools supporting KIF.

The technologies described in this section are invented for rule interchange. They provide different underlying logics e.g. FOL, deontic logic, modal logic, and

[1] See: `http://sourceforge.net/apps/mediawiki/rifle/index.php`
[2] See: `http://silk.semwebcentral.org`
[3] See: `http://code.google.com/p/fuxi`
[4] See: `http://www.upv.es/sma/teoria/sma/kqml_kif/kif.pdf`

support different inference modes. Despite the provided formalism, they cannot be applied to representation that are only a programming solution providing rule language. What is more, most of them are very general and thus become very complex and cannot be efficiently applied in a practical way. They work in a simple examples but in case of more complex rule bases they fail.

In the current stage, our research focuses on a systematic analysis of the classic rule languages providing forward chaining like CLIPS, Jess, Drools, etc. The analysis involves such elements like identification of underlying logics, inference modes, rule language syntax, tools support, etc. Before the thread discussed here, the analysis of the CLIPS rule language has been done [5]. In turn, this paper focuses on the Drools rule language and identification of a rule representation model features. In the contrast to described methods, thanks to our approach, among set of different rule representations we can select a subset for which efficient rule interoperability methods can be defined.

4 Multidimensional Approach to Rule Interoperability

This section provides a discussion concerning specification of the rule representation model. The specification tries to identify all the features of the considered rule representations, which should be supported by our rule representation model. In this paper, a rule base is considered from the static and dynamic perspectives. Through static perspective we refer to a rule language syntax that is used for specification of a rule base. In turn, dynamic perspective is related to logical interpretation of a knowledge base. This is an another approach to rule base analysis in comparison with our previous work, where the rule base was analyzed on four different levels: maintenance level, syntax level, semantics level and processing level [5]. Thanks to the previous approach, it was possible to identify the differences between XTT2 and other representations on all mentioned levels. The approach, that is described in this paper, allows for slightly different analysis of a rule base, because it allows for better identification of the features of other representations that should be supported by our model. What is more, considering knowledge interoperability methods, we must distinguish pure knowledge semantics from other elements having an impact on their semantics. Thanks to this approach, such distinction becomes easier and leads to identification of the *knowledge semantics invariants* i.e. features that do not affect a rule semantics directly but only through the inference process.

4.1 Dynamic Perspective

In this section a dynamic perspective is considered. As it was mentioned, this perspective concerns logical interpretation of rule bases. Many rule languages do not provide any logical interpretation, what in some situations may lead to ambiguity of language constructs. Below, an example of two rules (encoded in Drools Rule Language) is presented:

```
rule "rule1"
when
  Person(age != 100)
then
  // do some actions
end
```

```
rule "rule2"
when
  not(Person(age == 100))
then
  // do some actions
end
```

Consider is there ane difference between these two rules: Let assume that we have a set F containing all facts from fact base and a set S containing facts that determine if the above rules should be fired or not. For the first rule, the set S can be defined as follows:

$$S = \{f \in F : typeof(f) = Person \ \wedge age \neq 100\}$$

In turn, for the second rule, the set S can be defined as follows:

$$S = \{f \in F : typeof(f) = Person \ \wedge age = 100\}$$

Having defined the sets S for both rules, we can give an answer what are the differences between these two rules. The rule `rule1` is fired for every $f \in S$ i.e. the rule can be fired as many times as many facts are in the set S. This rule is not fired when $S = \emptyset$. In turn, the rule `rule2` is fired only when the $S = \emptyset$ i.e. there are no facts of the type `Person` having a value of `age` equal to 100 at all. The goal of the above example is to show how important a precisely defined semantics of a rule language is. Moreover, how important it is to provide, in the context of dynamic perspective, a logical interpretation of a language.

An expressiveness of the considered rule languages can be covered by the first order logic. However, they also provide many concepts that are known from modern programming languages like: data types, operators, inheritance, etc.

Data Types. Nowadays, the Object Oriented Programming (OOP) languages becomes very common. This is because they allow for referring to different elements as to objects what is very intuitive and easy to understand. The OOP paradigm is also ported to rule languages. The example of such combinations is Drools which is strongly connected with JAVA language. The Drools Rule Language provides type system based on JAVA data types. It is possible to define facts that are integers or strings. On the other hand, the First Order Logic does not provide any type system. Thus, at once first problem appears: There is no logic-based definition what does it mean that a fact `f` is an instance of type `F`.

Complex Types. In the OOP paradigm, a single object can be a composition of many factors that are used to describe it. For example, a object `car` can be described by `color`, `capacity`, `power`, etc. In addition, common OOP programming languages allow for defining functions/methods that are related to a given type. The OOP provided by rule languages usually allows for defining a composition of the factors as a single type but does not support defining methods.

Additionally, rule languages provide two classes of complex types: ordered types and unordered types. The unordered ones correspond to those known from OOP, where each factor is identified by name. In turn, in ordered types, factors do not have a name but are identified by index. The example below provides definitions of two types describing a person by using `name`, `age` and `address`.

```
declare unordered_type_of_Person
    name : String
    age : Integer
    address : String
end
```

and

```
(ordered_type_of_Person name age address)
```

Order of field definitions while instantiation of unordered types is unimportant:

```
unordered_type_of_Person up
up.name = "Kris"
up.address = "Some address"
up.age = 18
```

is equivalent to:

```
unordered_type_of_Person up
up.age = 18
up.name = "Kris"
up.address = "Some address"
```

In the case of ordered types, order of fields definitions is important, e.g.:

```
(ordered_type_of_Person
   "Kris" 18 "Some address")
```

is not the same as

```
(ordered_type_of_Person
   18 "Kris" "Some address")
```

All issues related to complex types should also be formalized in order to define an expressiveness of such definitions.

Operators. The type system is strongly related to operators and determines set and behavior of the operators that can be applied for a given type. For example, lets assume that we have a set of all facts $F = \{f_{i1} = 7, f_{i2} = 10, f_{s1} = 7, f_{s2} = 10\}$, where the type of f_{i1} and f_{i2} is integer, while the type of f_{s1} and f_{s2} is string. Performing operations provided by operators $<$ or $+$ we have:

$$f_{i1} < f_{i2} = true \quad \text{while} \quad f_{s1} < f_{s2} = false$$
$$f_{i1} + f_{i2} = 17 \quad\quad\quad\quad f_{s1} + f_{s2} = 710$$

Considering above example, definition of operators‘ semantics behavior is a must.

The other issue that is connected to operators is related to usage context. There an two different usage contexts for operators corresponding to the conditional and conclusion part of a rule. Some of the operators can be used only in one context while others in both. The formalization must precisely define limitations of operator usage and its semantics in a given context.

Inheritance. CLIPS does not provide an inheritance mechanism in core part of the language. However, it provides a COOL (Clips Object Oriented Language) syntax which supports most of the OOP features including inheritance. In turn, Drools provides inheritance natively. However, Drools mechanism does not provide complete support for all features related to inheritance. It only allows for extending a new type by already declared base type. The fact instances of the extended type contain all fields from a base type. They can also be interpreted as an instances of the base type.

The inheritance mechanism plays important role in the *pattern-matching* step of inference algorithm. Drools provides a dedicated version of the classic Rete algorithm [19] called ReteOO which supports inference about objects. Lets assume that we have a fact type `Person` which is extended by type `Student` (each student is a person). Each modification of the fact of type `Students` affects activations of rules containing facts of type `Person` in their conditional part.

Such behavior may seem obvious, however it cannot be assumed that the inheritance mechanism works always in this way. Use of derived types in the

context of programming languages involves many other aspects especially such as polymorphism, abstractions, etc. It also cannot be assumed that all the inheritance features are supported. Thus, the knowledge representation model must precisely define to what degree the inheritance is supported.

Set of Values. The XTT2 formalization [4] provides a concept of generalized attributes which can take a set as a value. A similar mechanism is also supported by CLIPS with the help of multifield slots [5]. Because of the strong connection with JAVA, Drools can also provide a similar concept by defining field as some collection (array, list, vector, etc).

Nevertheless, there are differences between these concepts. Some of them are ordered sets while some of them are unordered sets. This two types of sets must be distinguished in order to define a set of possible operators that can be used.

Domains. Nowadays, it can be noticed that two aspects of the rule bases modeling have increased: 1) the average number of rules in a rule base and 2) the emphasize on rule base quality. It is obvious that the more rules are in a rule base the more difficult is to ensure a high quality of rule base. This is why, different tools for rule bases verification are developed. They work on different levels starting from syntax level where a rule base is analyzed against syntax errors to logical level where a rule base is analyzed against many different logical anomalies like consistency, determinism, contradictions, etc. In order to provide an efficient tool for detecting logical anomalies, a set of possible values of factors must be defined. Such set of values is defined as a domain.

Definition of domain concept cannot be introduced separately because it affects other definitions. For example, lets assume that a rule base contains a fact f_i of the type integer. Then, the conditional expression $f_i < 7$ is satisfied by values from interval $(-\infty; 7)$. When a domain for this fact is defined as the interval $[0; 100]$, then the above conditional expression is satisfied by values from interval $[0; 7)$. This is result of the intersection of two intervals: $(-\infty; 7) \cap [0; 100]$.

The other issue, that extends the concept of domains, is the data integrity. The concept of domain is related to a single factor. It is also possible to define the dependencies between values of two or more factors. For example, lets assume the type `Person` is described by factors `age` and `isAdult`. The domains for each factor are defined respectively: $[0; 150]$, $\{false, true\}$. Nevertheless, the values of these fields are depended and the value of `isAdult` can be equal to `true` only when the value of `age` ≥ 18. In other cases, the value of `isAdult` must be equal to `false`. Business Rules provide a special type of a rule called *integrity rule* that is used for ensuring data integrity is such cases.

4.2 Static Perspective

The previous section was focused on the dynamic aspects of the rule base. This section refers to static issues that are related to rule language syntax and expressiveness. It does not provide a complete specification of all issues, but discusses only the most important ones.

It is worth to notice that the model, which is subject of our research, involves also definition of the dedicated rule language. This is why, issues discussed in Section 4.1 also have impact on the static aspects, since a rule language must provide appropriate syntax for them. The issues discussed in this section cannot be considered separately, because definition of their semantics should also be provided.

Rule Base Modularization. Most of the rule languages provide some mechanisms allowing for organizing the rule bases. Usually, it allows for grouping rules within so-called modules. Placing a rule into a module brings different effects in different representations [20]. It mainly has an impact on the inference process. Because of the fact, that a rule base modularization cannot be treated as neutral in the context of knowledge, it support cannot be omitted in the model.

Facts Binding. Facts binding is a very important mechanism that must be supported by rule language. It allows for performing operations on the facts that satisfy a rule. While a rule is analyzed (the conditional part is checked against facts), reference to matched facts may be stored into a local rule variables. Thanks to such references, matched facts becomes available and can be use as normal objects. The following example presents a rule that modifies all facts of the type `Person` by changing value of factor `isAdult` to `true` when a given person is at least 18 years old. The rule uses a variable `p` for storing fact that satisfies their conditional part:

```
rule "updatePerson"
when
  p : Person(age >= 18)
then
  p.isAdult = true
end
```

Because, this is only mechanism allowing for such operations, it must be supported by the syntax of a rule language.

Variables. The considered rule languages provide a concept of variables. Similarly to facts, variables allow for storing some information. However, in contrast to facts, variables can be treated as non knowledge-based elements that do not affect inference process. They are usually used for storing values of constants, threshold, factors, etc. Thanks to its nature, they allow for creating more clear rules and facilitate their modifications.

4.3 Knowledge Semantics Invariants

In the context of rule interoperability, analysis presented in this paper aims at identifying features of the rule representations that should be supported by our model. This analysis involves two aspects of the rule bases: statical and dynamical. Thanks to this approach, an another aspect of a knowledge base can be distinguished. This aspect is related to features that cannot be considered as knowledge-based elements, however, they may affect knowledge semantics while processing. These elements are usually referred as rule properties and allow for providing hints for inference algorithm how to process knowledge.

5 Summary and Future Work

This paper presents work in progress research aiming at providing efficient rule interoperability method. Our work does not assume support for all rule representation languages and formalisms but for a well defined subset. This will prevent our method from being very general and thus impractical.

The most important stage of our research is development of a rule representation model covering an expressiveness of all rule languages in the selected subset. Definition of the model involves two dimensions of the rule base: formalization of a rule base and definition of a dedicated rule language. Our starting point for this stage is the formalization provided by XTT2 method [4]. However, analysis of the selected rule languages must be performed before the model definition. The goal of this analysis is to compare the considered rule languages against XTT2 in order to identify their limitations. This part of our work is described in [5]. The second step in this stage involves identification of the model features that must be take into account. This step is the main subject of this paper and the provided discussion is the original contribution for our research.

The future works cover realization of the next steps towards definition of the efficient rule interoperability methods. These step include a development of the model formalization, which can be done thanks to the identified limitations and required features. Then, a dedicated rule language can be provided.

All considered rule representations provide only rule language without underlying logic interpretation. This is why, during the subsequent step the logical interpretation of these languages will be provided with the help of the model. At the end of this step, all the considered languages will have a unified logical interpretation. In turn, this will facilitate the definition of rule interoperability methods — the differences between languages will be visible very clearly. What is more, limitations of the methods will be described in formal way.

The last stage of our work involves providing tool support. We assume to develop a tool supporting the expressiveness and rule language provided by the model. This tool will allow for modeling rule bases and then exporting them into supported representations. Translation in opposite direction will be possible thanks to an import tools. In our previous works, we already developed knowledge translators that are based on a XSLT processor [21].

References

1. Ligęza, A.: Logical Foundations for Rule-Based Systems. Springer (2006)
2. Giurca, A., Gašević, D., Taveter, K. (eds.): Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches. Information Science Reference, New York (2009)
3. Ligęza, A., Nalepa, G.J.: A study of methodological issues in design and development of rule-based systems: proposal of a new approach. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1(2), 117–137 (2011), http://onlinelibrary.wiley.com/doi/10.1002/widm.11/pdf
4. Nalepa, G.J., Ligęza, A., Kaczor, K.: Formalization and modeling of rules using the XTT2 method. International Journal on Artificial Intelligence Tools 20(6), 1107–1125 (2011)

5. Kaczor, K., Nalepa, G.J.: Critical evaluation of the XTT2 rule representation through comparison with CLIPS. In: Canadas, J., Nalepa, G.J., Baumeister, J. (eds.) 8th Workshop on Knowledge Engineering and Software Engineering (KESE2012) at the at the Biennial European Conference on Artificial Intelligence (ECAI 2012), Montpellier, France, August 28 (2012), http://ceur-ws.org/Vol-949/
6. Nalepa, G.J.: PlWiki – A generic semantic wiki architecture. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS, vol. 5796, pp. 345–356. Springer, Heidelberg (2009)
7. OMG, Semantics of Business Vocabulary and Business Rules (SBVR). Object Management Group, Tech. Rep. dtc/06-03-02 (2006)
8. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S.: OWL 2 Web Ontology Language – primer. W3C, W3C Recommendation (October 2009)
9. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
10. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A semantic web rule language combining OWL and RuleML, W3C member submission 21 may 2004. W3C, Tech. Rep (2004)
11. Wagner, G.: How to design a general rule markup language. In: XML Technology for the Semantic Web (XSW 2002), HU Berlin. Lecture Notes in Informatics, pp. 19–37 (2002)
12. Adrian, W.T., Nalepa, G.J., Kaczor, K., Noga, M.: Overview of selected approaches to rule representation on the Semantic Web. AGH University of Science and Technology, Tech. Rep. CSLTR 2/2010 (2010), http://cslab.ia.agh.edu.pl/csltr:listaraportow
13. Nalepa, G.J.: Semantic Knowledge Engineering. A Rule-Based Approach. Wydawnictwa AGH, Kraków (2011)
14. Fensel, D., Angele, J., Landes, D.: KARL: a knowledge acquisition and representation language. In: Rault, J.C. (ed.) Proceedings of the 11th International Conference Expert Systems and their Applications. Tools, Techniques & Methods, vol. 1, pp. 513–528. EC2, Nanterre, Avignon (1991)
15. van Harmelen, F., Balder, J. (ML)2: A formal language for KADS models. In: ECAI, pp. 582–586 (1992)
16. Kifer, M., Boley, H.: RIF overview. W3C, W3C Working Draft (October 2009), http://www.w3.org/TR/rif-overview
17. Boley, H., Tabet, S., Wagner, G.: Design rationale for ruleml: A markup language for semantic web rules. In: Cruz, I.F., Decker, S., Euzenat, J., McGuinness, D.L. (eds.) SWWS, pp. 381–401 (2001)
18. Wagner, G., Giurca, A., Lukichev, S.: R2ml: A general approach for marking up rules. In: Bry, F., Fages, F., Marchiori, M., Ohlbach, H. (eds.) Principles and Practices of Semantic Web Reasoning, Dagstuhl Seminar Proceedings 05371 (2005)
19. Forgy, C.: Rete: A fast algorithm for the many patterns/many objects match problem. Artif. Intell. 19(1), 17–37 (1982)
20. Nalepa, G., Bobek, S., Ligęza, A., Kaczor, K.: Algorithms for rule inference in modularized rule bases. In: Bassiliades, N., Governatori, G., Paschke, A. (eds.) RuleML 2011 - Europe. LNCS, vol. 6826, pp. 305–312. Springer, Heidelberg (2011)
21. Nalepa, G.J., Kluza, K.: UML representation for rule-based application models with XTT2-based business rules. International Journal of Software Engineering and Knowledge Engineering (IJSEKE) 22(4), 485–524 (2012), http://www.worldscientific.com/doi/abs/10.1142/S021819401250012X

A New Approach to Designing Interpretable Models of Dynamic Systems

Krystian Łapa, Andrzej Przybył, and Krzysztof Cpałka

Częstochowa University of Technology,
Institute of Computational Intelligence, Poland
{krystian.lapa,andrzej.przybyl,krzysztof.cpalka}@iisi.pcz.pl

Abstract. In the process of designing automatic control system it is very important to have an accurate model of the controlled process. Approaches to modelling dynamic systems presented in the literature are often approximate, uninterpretable (acting as a black box), not appropriate to work in real-time, so it is not possible to create a hardware emulator on the basis of these approaches. The paper presents a new method to create model of nonlinear dynamic systems which gives a real opportunity for the interpretation of accumulated knowledge. By combining methods of control theory with fuzzy logic rules a good accuracy of the model can be achieved with use of a small number of fuzzy rules. Our method is based on the evolutionary strategy (μ, λ).

1 Introduction

Modelling of the systems is a widely developed area. In the literature many topics connected with modelling issue are considered. In the last years, besides classic solutions, modelling solutions based on the artificial neural networks and fuzzy logic rules are presented in e.g. [11], [16], [21]-[24], [31]-[34], [39]. It should be noted that most of the papers which use above-mentioned methods relate to the phenomena that can be simply described by "input-output" type transposition

$$\mathbf{y} = F(\mathbf{u}), \tag{1}$$

where $\mathbf{y}$ and $\mathbf{u}$ are vectors of input and output signals. In reality most of physical phenomena are dynamic and state of them is not only depended on input of current signals, but also on their prior states. Inclusion of historical data in the input vector $\mathbf{u}$ (i.e. prior values of inputs and/or outputs) enables consideration of dynamic dependences in the designed model. However, in the general case that model may be too complex and uninterpretable, what makes it usefulness in the practice. This disadvantage is a characteristic of modelling by artificial neural networks and also modelling based on the fuzzy rules which input vector was enlarged by the historic data. Another way of modelling is the use of the state variables technique. State of the dynamic object model may be comprehensively described by vector of state variables which has an appropriate size [17]. Vector of state variables describes completely a state of the object, what means that

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 523–534, 2013.

knowledge about it at time t, and knowledge about input signals (vector $\mathbf{u}$) at the next moments, with known model of the object, gives a complete knowledge about the object behaviour. The model in a linear case is described as follows

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{2}$$

where $\mathbf{A}$ and $\mathbf{B}$ are system output matrices of an appropriate size. In the non-stationary models a coefficients of matrix which describe the model change in the time function. In the stationary models the coefficients are constant. In a more general case - i.e. for non-linear objects, equation (2) takes a form

$$\frac{d\mathbf{x}}{dt} = F\left(\mathbf{x}, \mathbf{u}\right), \tag{3}$$

where F is a non-linear dependency in the function of state variables and input signals. Modelling of the non-linear objects and phenomena is much more complicated. However, it should be noted that many physical phenomena may be described by local linear approximation (2) of non-linear dependency (3) about an operating point [17], [28]. Operating point changes over time during the process. However, a local re-determination of linear approximation in any new point is possible. For the discretization with the suitable short time step T that solution is enough accurate, even if the first order approximation is used, i.e.

$$\mathbf{x}\left(k+1\right) = \mathbf{I} + \mathbf{A}\left(k\right)\mathbf{x}\left(k\right) \cdot T + \mathbf{B}\left(k\right)\mathbf{u}\left(k\right) \cdot T, \tag{4}$$

where $\mathbf{I}$ is the identity matrix with the appropriate size. Designed models will refer to continuous objects noted as discrete form with time step T, connected with the current time t by the dependency $t = kT$, where $k = 1, 2, ...$ Modelling with use of the dynamic phenomena description as state variables and fuzzy rules will be based on the canonical form of the state equations [17] how it was explained in the previous paper [28].

The idea of our method consist in taking advantage of computational intelligence in the modelling of nonlinear systems. In the commonly used modelling method which models the industrial nonlinear dynamic systems, the applied intelligent systems were used as a "black box" or as a "grey box" models. It should be noted that the systems in many cases replaced commonly used and reputable solutions from the classic control theory. The authors would like to create a new hybrid method which will be used to support that solutions from control theory, but it will also make available the knowledge which describes a mechanism of the model work. The task of our intelligent systems is not modelling input-output dependencies. In this approach a large number of rules are required. This implies that the rules become illegible. In our approach an intelligent system is used to generate the coefficients of the matrices of the state-vector equation. For learning the intelligent system we use an evolutionary strategy (μ, λ) and data obtained from the observation of the real object.

The proposed approach has not yet been described in the literature by other authors. In paper [27] only a little similar conception of the authors has been

presented. The conception was successfully used in the project of adaptive observer of state variables of induction motor. Received results confirm very well qualities of this solution. The results confirm also rightness of the presumptions of new algorithm development. In a previous paper [28] we simplified considered the problem, i.e. we assumed that we know the values of some coefficients of the system output matrices (i.e. matrix **A**). In this paper we continue the investigations, however, we analyse the more generalized case.

This paper is organized into six sections. In the next section an idea of the proposed modelling method is presented. In Section 3 we describe neuro-fuzzy system for nonlinear modelling. Section 4 shows the evolutionary generation of the interpretable models of dynamic systems and Section 5 presents experimental results. Conclusions are drawn in Section 6.

2 Idea of the Proposed Modelling Method

Comments on the proposed method can be summarized as follows:

- Our method allows to use of a lot of methods designed for analysis and design of linear model, also in use of the non-linear objects.
- Our method is based on the both described techniques: numeric modelling by the local linear approximation (about an current operating point) of nonlinear object and fuzzy rules allow to practical interpretation of new received knowledge. In the proposed solution obtainment of accurate model of nonlinear object (performed by fuzzy rules) and analysis of the model by techniques developed for linear models are possible, so the solution allows on automatic building of the object model based on the fuzzy rules.
- Our method based on the fuzzy rules enables interpretation of knowledge which was automatically obtained by an experiment during the observation of the real object. Interpretability of the model of any object is the result of the knowledge obtainment possibility about physical processes of modelled object. This is an attribute of the objects based on the fuzzy rules which are described as linguistic form.
- Our method (based on the algebraic equations with support of small number of fuzzy rules) allows for hardware realization, e.g. in the FPGA structures, and allows to real use of high potential of the novel soft-computing methods.

The approach proposed by us is illustrated in the Fig. 1 and uses fuzzy rules to generate the coefficients of matrices which are in algebraic notation of equations (3). The coefficients define the model. If the model is non-linear or non-stationary, the coefficients of matrices will change over time function or selected state variables function. The dependency between selected matrix coefficients will be described by the fuzzy rules in an interpretable way (Fig. 1). Form and number of the fuzzy rules will be automatically selected in the procedure of minimization of properly defined objective function with use of the evolutionary algorithm. If the approximate linear model is available, it will be used as initial model which will be improved in the next steps the of algorithm.

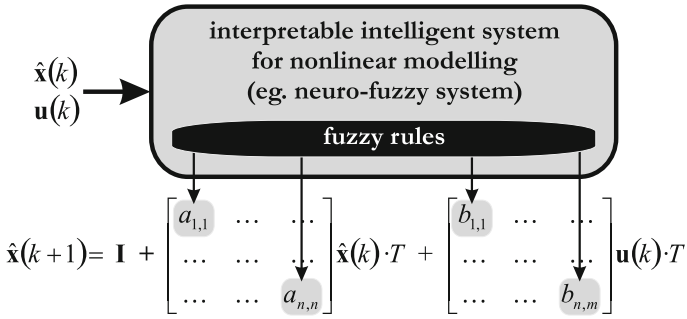


Fig. 1. Idea of the modelling method based on the fuzzy logic rules and modelling technique with use of dynamic state object variables

3 Neuro-Fuzzy System for Nonlinear Modelling

We propose the neuro-fuzzy system for nonlinear modelling, because the knowledge contained in it is interpretable. In literature various neuro-fuzzy systems have been developed (see e.g. [4], [9], [10], [15], [19]-[26], [30], [34]-[41]). They combine the natural language description of fuzzy systems and the learning properties of neural networks (see e.g. [1]-[3], [5]-[8], [33]).

We consider a multi-input, multi-output neuro-fuzzy system, mapping $\mathbf{X} \to \mathbf{Y}$, where $\mathbf{X} \subset \mathbf{R}^n$ and $\mathbf{Y} \subset \mathbf{R}^m$. The fuzzy rule base of the system consists of a collection of N fuzzy IF-THEN rules in the form

$$R^k : \left[\text{IF } x_1 \text{ is } A_1^k \text{ AND} \dots \text{AND } x_n \text{ is } A_n^k \text{ THEN } y_1 \text{ is } B_1^k, \dots, y_m \text{ is } B_m^k \right], \tag{5}$$

where $\mathbf{x} = [x_1, \dots, x_n] \in \mathbf{X}$, $\mathbf{y} = [y_1, \dots, y_m] \in \mathbf{Y}$, $A_1^k, A_2^k, \dots, A_n^k$ are fuzzy sets characterized by membership functions $\mu_{A_i^k}(x_i)$, $i = 1, \dots, n$, $k = 1, \dots, N$, whereas B_j^k are fuzzy sets characterized by membership functions $\mu_{B_j^k}(y_j)$, $j = 1, \dots, m$, $k = 1, \dots, N$.

Each of N rules (5) determines fuzzy sets $\bar{B}_j^k \subset \mathbf{Y}$ given by

$$\mu_{\bar{B}_j^k}(y_j) = \mu_{A_1^k \times \dots \times A_n^k \to B_j^k}(\bar{\mathbf{x}}, y_j) = \mu_{\mathbf{A}^k \to B_j^k}(\bar{\mathbf{x}}, y_j) = T\left\{ \tau_k(\bar{\mathbf{x}}), \mu_{B_j^k}(y_j) \right\}, \tag{6}$$

where $T\{\}$ is a t-norm (see e.g. [18]) and $T\left\{ \tau_k(\bar{\mathbf{x}}), \mu_{B_j^k}(y_j) \right\}$ denotes an inference operator in the Mamdani-type system (see e.g. [34]). As a result of aggregation of the fuzzy sets $\bar{B}_j^k$, we get set B_j' with membership function given by

$$\mu_{B_j'}(y_j) = \underset{k=1}{\overset{N}{S}} \left\{ \mu_{\bar{B}_j^k}(y_j) \right\}. \tag{7}$$

The defuzzification is realized by the COA method defined by the following formula

$$\bar{y}_j = \frac{\sum\limits_{r=1}^{N} \bar{y}_{j,r}^B \cdot \mu_{B'_j}\left(\bar{y}_{j,r}^B\right)}{\sum\limits_{r=1}^{N} \mu_{B'_j}\left(\bar{y}_{j,r}^B\right)}, \tag{8}$$

where $\bar{y}_{j,r}^B$ are centers of the membership functions $\mu_{B_j^r}(y_j)$, i.e. for $j = 1, \ldots, m$ and $r = 1, \ldots, N$, we have

$$\mu_{B_j^r}\left(\bar{y}_{j,r}^B\right) = \max_{y_j \in \mathbf{Y}} \left\{\mu_{B_j^r}(y_j)\right\}. \tag{9}$$

In our investigation we used the neuro-fuzzy system (8) to produce the values of coefficients of matrix $\mathbf{A}$ of the equation (3) as shown in the Fig. 1. Detailed description of the system (8) can be found in [10].

4 Evolutionary Generation of the Interpretable Models of Dynamic Systems

We used the evolutionary strategy (μ, λ) in the process of creating the interpretable model of the dynamic systems. The purpose of this is to obtain the parameters of neuro-fuzzy system described in the previous section. In the process of evolution we assumed that:

- In a single chromosome $\mathbf{X}_{ch}^{\text{par}}$, according to the Pittsburgh approach, a complete linguistic model is coded in the following way

$$\begin{aligned}\mathbf{X}_{ch}^{\text{par}} &= \begin{pmatrix} \bar{x}_{1,1}^A, \sigma_{1,1}^A, \ldots, \bar{x}_{n,1}^A, \sigma_{n,1}^A, \bar{y}_{1,1}^B, \sigma_{1,1}^B, \ldots, \bar{y}_{m,1}^B, \sigma_{m,1}^B, \ldots \\ \bar{x}_{1,N}^A, \sigma_{1,N}^A, \ldots, \bar{x}_{n,N}^A, \sigma_{n,N}^A, \bar{y}_{1,N}^B, \sigma_{1,N}^B, \ldots, \bar{y}_{m,N}^B, \sigma_{m,N}^B \end{pmatrix}, \\ &= \left(X_{ch,1}^{\text{par}}, X_{ch,2}^{\text{par}}, \ldots, X_{chL^{\text{par}}}^{\text{par}}\right)\end{aligned} \tag{10}$$

  where $ch = 1, \ldots, \mu$ for parent population or $ch = 1, \ldots, \lambda$ for the temporary population and $L^{\text{par}} = 2 \cdot N \cdot (n + m)$ denotes length of the chromosome (10).
- In a single binary chromosome $\mathbf{X}_{ch}^{\text{red}}$ is coded information about which the input and output fuzzy sets and the rules in the system (8) are redundant (their reduction does not affect the accuracy of the system (8))

$$\begin{aligned}\mathbf{X}_{ch}^{\text{red}} &= \begin{pmatrix} A_1^1, \ldots, A_n^1, B_1^1, \ldots, B_m^1, \text{rule}_1, \ldots, \\ A_1^N, \ldots, A_n^N, B_1^N, \ldots, B_m^N, \text{rule}_N \end{pmatrix}, \\ &= \left(X_{ch,1}^{\text{red}}, X_{ch,2}^{\text{red}}, \ldots, X_{ch,L^{\text{red}}}^{\text{red}}\right)\end{aligned} \tag{11}$$

  where $ch = 1, \ldots, \mu$ for parent population or $ch = 1, \ldots, \lambda$ for the temporary population and $L^{\text{red}} = N \cdot (n + m + 1)$ denotes length of the chromosome (11). The genes in the chromosome $\mathbf{X}_{ch}^{\text{red}}$ take values from the set $\{0, 1\}$.

– Each neuro-fuzzy system has a number of outputs equal to the number of matrix $\mathbf{A}$ coefficients. Fitness function is based on the differences between output signals $\hat{\mathbf{x}}(\cdot)$ generated by the created model at step $k+1$ and corresponding reference values $\mathbf{x}(\cdot)$. Starting values for the model are the reference values at step k

$$\mathrm{ff}(\mathbf{X}_{ch}) = \sqrt{\frac{1}{2Z}\sum_{z=1}^{Z}\begin{pmatrix}(x_1(z+1)-\hat{x}_1(z+1))^2+\\(x_2(z+1)-\hat{x}_2(z+1))^2\end{pmatrix}}. \tag{12}$$

– Genes in chromosome $\mathbf{X}_{ch}^{\mathrm{par}}$ which correspond to the input fuzzy sets ${A_i}^k$, $k=1,\ldots,N$, $i=1,\ldots,n$, ($\bar{x}_{i,k}^A$ and $\sigma_{i,k}^A$) and genes which correspond to the output fuzzy sets B_j^k, $k=1,\ldots,N$, $j=1,\ldots,m$ ($\bar{y}_{j,k}^B$ and $\sigma_{j,k}^B$) were initialized on the basis of the method described in [10] and [14].
– Genes in chromosome $\mathbf{X}_{ch}^{\mathrm{red}}$ were chosen as random numbers.

For more details on the evolutionary strategy (μ,λ) used to modify the parameters and structure of the neuro-fuzzy system (8) please see [13], [14], [33].

5 Experimental Results

In our work we considered the van der Pol oscillator [42] which is used in the medicine as the model of the heartbeat. We have attempted to identify that model on the basis of the reference data which were generated using the adequate differential equation

$$\frac{d^2x}{dt^2}+\alpha\left(x^2-1\right)\frac{dx}{dt}+\omega^2x=0, \tag{13}$$

where α, ω are oscillator parameters and $x(t)$ is a reference value of the modelled process as a function of time (in the simulations we assumed that $\alpha=10$ and $\omega=1$). We used the following state variables: $x_1(t)=x(t)$, $x_2(t)=dx(t)/dt$. In such a case the system matrix $\mathbf{A}$ described by the formula (2) takes the following form

$$\mathbf{A}=\begin{bmatrix}0 & 1\\ -\omega^2 & -\alpha\left(x_1^2-1\right)\end{bmatrix}=\begin{bmatrix}a_{11}(\mathbf{x}) & a_{12}(\mathbf{x})\\ a_{21}(\mathbf{x}) & a_{22}(\mathbf{x})\end{bmatrix}. \tag{14}$$

The goal of the modelling was to recreate the unknown parameters $a_{11}(\mathbf{x})$, $a_{12}(\mathbf{x})$, $a_{21}(\mathbf{x})$, $a_{22}(\mathbf{x})$ in such a way that the model reproduces the reference data as accurately as possible. Of course in a general case analytical dependencies which were used to generate the reference data are not known. However, the proposed method allows us to reconstruct these dependencies in the form of a set of interpretable fuzzy rules (5). This is possible on the basis of the analysis of measurable outputs of the modelled process. In our case the measurable output signals are $x_1(k)$ and $x_2(k)$. They are used as reference values for the outputs. Output signals generated by the created model $\hat{x}_1(k)$ and $\hat{x}_2(k)$ are compared with their reference values and the error of the model is calculated (12).

Details of the simulations can be summarized as follows:

- The neuro-fuzzy system (8) used in the simulations is characterized by Gaussian membership functions, min/max triangular norms, $N = 3$ and $m = 4$.
- The evolutionary strategy (μ, λ) used for the learning of the intelligent system (8) is characterized by the following parameters: $\mu = 10$, $\lambda = 500$, $p_m = 0.077$, $p_c = 0.770$, and the number of generations = 100000 (for details see e.g. [10]).

The results of simulations, depicted in Table 1 and presented in the Fig. 3 and Fig. 4, can be summarized as follows:

- Neuro-fuzzy system obtained in evolutionary learning is characterized by two rules ($N = 2$), two inputs ($\hat{x}_1(k)$ and $\hat{x}_2(k)$) and four outputs ($a_{11}(k)$, $a_{12}(k)$, $a_{21}(k)$ and $a_{22}(k)$).
- A small number of rules in the obtained system allows on their interpretation. Equation (5) can be written as follows

$$\begin{cases} R^1 : \left[\text{IF} x_1 \text{is} A_1^1 \text{AND} x_2 \text{is} A_2^1 \text{THEN} y_1 \text{is} B_1^1, y_3 \text{is} B_3^1, y_4 \text{is} B_4^1 \right] \\ R^2 : \left[\text{IF} x_1 \text{is} A_1^2 \text{AND} x_2 \text{is} A_2^2 \text{THEN} y_2 \text{is} B_2^2, y_4 \text{is} B_4^2 \right] \end{cases} . \tag{15}$$

Equation (15) can be written as follows

$$\begin{cases} R^1 : \left[\begin{array}{l} \text{IF} x_1(k) \text{is} \mathbf{low} \text{AND} x_2(k) \text{is} \mathbf{low} \\ \text{THEN} a_{11}(\mathbf{x}) \text{is} \mathbf{medium}, a_{21}(\mathbf{x}) \text{is} \mathbf{medium}, a_{22}(\mathbf{x}) \text{is} \mathbf{low} \end{array} \right] \\ R^2 : \left[\begin{array}{l} \text{IF} x_1(k) \text{is} \mathbf{high} \text{AND} x_2(k) \text{is} \mathbf{high} \\ \text{THEN} a_{12}(\mathbf{x}) \text{is} \mathbf{medium}, a_{22}(\mathbf{x}) \text{is} \mathbf{high} \end{array} \right] \end{cases} , \tag{16}$$

where "low", "medium" and "high" are the values of the linguistic variables $x_1(k)$, $x_2(k)$, $a_{11}(\mathbf{x})$, $a_{12}(\mathbf{x})$, $a_{21}(\mathbf{x})$ and $a_{22}(\mathbf{x})$ and are represented by fuzzy sets shown in the Fig. 2.

- The coefficients of the matrix $\mathbf{A}$, generated by neuro-fuzzy system (8) as a function of $x_1(k)$ and $x_2(k)$ are presented in the Fig. 3. We can see that the values of the coefficient $a_{11}(\mathbf{x})$ takes the value like 0 and $a_{12}(\mathbf{x})$ takes the value like 1. We can also see that the values of $a_{21}(\mathbf{x})$ and $a_{22}(\mathbf{x})$ vary according to the theoretical model described by (13) and (14).
- Accuracy of the nonlinear modelling obtained in simulations is shown in Table 1 and presented in the Fig. 4. The evolutionarily selected neuro-fuzzy system (8) works with good accuracy. It should be emphasized that our goal was not obtaining the best performance of neuro-fuzzy systems, but we wanted to show that it is possible to automatically design neuro-fuzzy structures and to find all parameters of such structures characterized by a good performance. In addition, our goal was to show that our method of determining the coefficients of the matrix $\mathbf{A}$ can be described by transparent rules. These goals have been fully achieved.
- In the simulations a simple approach to nonlinear modelling were also considered. In this approach, the system (8) was directly used for modelling according to formula (2). The value of RMSE was equal 0.437 for the three rules ($N = 3$). Thus, modelling accuracy was lower compared to that obtained using the method proposed in this paper.

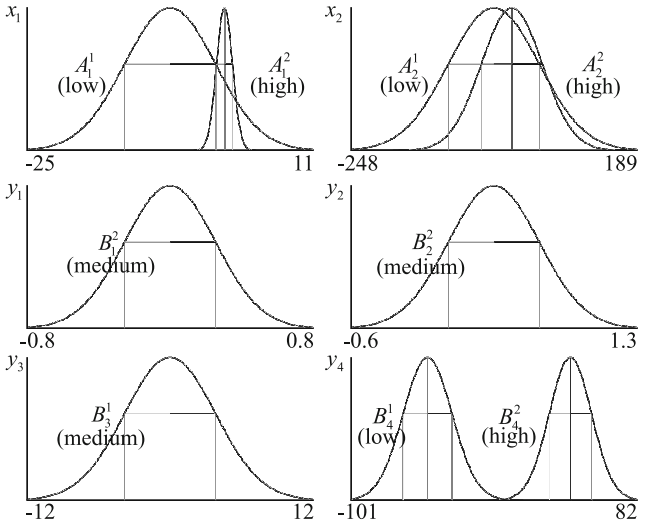


Fig. 2. Input and output fuzzy sets of the neuro-fuzzy system (8)

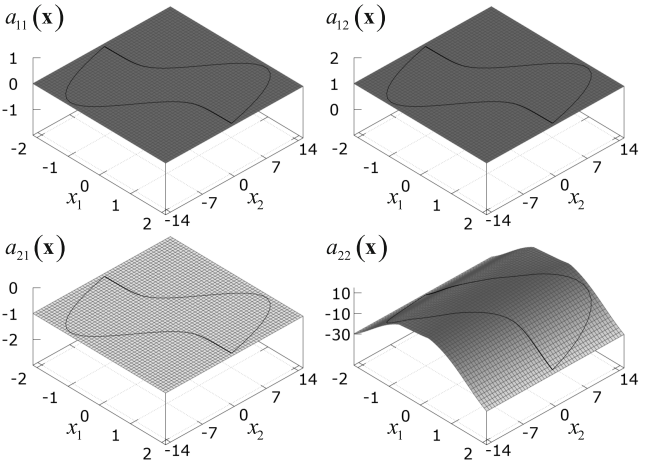


Fig. 3. The coefficients of the matrix **A**, generated by neuro-fuzzy system (8) as a function of $x_1(k)$ and $x_2(k)$

6 Summary

In the paper a new method to create model of nonlinear dynamic systems was proposed. The novelty was the combination of the method of control theory and fuzzy rules, which enables to obtain good accuracy of the model using a small number of fuzzy rules. Our method will be able to work in the real-time and it will be enough accurate for mapping in details many nonlinear dynamic systems,

Table 1. Accuracy of the nonlinear modelling obtained in simulations

Method	Number of rules	Average RMSE
A. Przybył, K. Cpałka ([28])	$N = 5$	0.0007
A. Przybył, K. Cpałka ([28])	$N = 3$	0.0035
our result	$N = 2$	**0.0145**

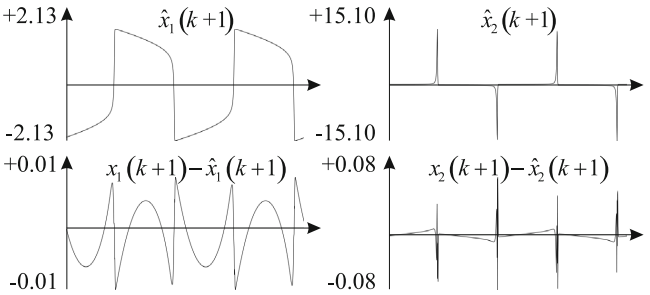


Fig. 4. Output and error signals obtained in the simulations

e.g. industrial processes or natural phenomena. It will be able to obtain the model by the non-invasive observation, transparent for the monitored process. In a case of industrial process modelling, data will be collected by the analysis of the packets which are sent in the real-time in the Ethernet network (see e.g. [29]). The proposed method is based on the evolutionary strategy (μ, λ) and allows to find both the structure and parameters of the used neuro-fuzzy system in the process of evolution. That model gives the potential possibility to the interpretation of accumulated knowledge. The simulation shows the fully usefulness of the proposed method.

Acknowledgment. The project was financed by the National Science Center on the basis of the decision number DEC-2012/05/B/ST7/02138.

References

1. Bartczuk, Ł., Rutkowska, D.: A new version of the fuzzy-ID3 algorithm. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 1060–1070. Springer, Heidelberg (2006)
2. Bartczuk, Ł., Rutkowska, D.: Medical diagnosis with type-2 fuzzy decision trees. Computers in Medical Activity, Advances in Intelligent and Soft Computing 65, 11–21 (2009)
3. Bartczuk, Ł., Rutkowska, D.: Type-2 fuzzy decision trees. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 197–206. Springer, Heidelberg (2008)

4. Casillas, J., Cordon, O., Herrera, F., Magdalena, L. (eds.): Interpretability Issues in Fuzzy Modelling. Springer (2003)
5. Cierniak, R.: A new approach to image reconstruction from projections problem using a recurrent neural network. Applied Mathematics and Computer Science 18(2), 147–157 (2008)
6. Cierniak, R.: A novel approach to image reconstruction problem from fan-beam projections using recurrent neural network. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 752–761. Springer, Heidelberg (2008)
7. Cierniak, R.: An image compression algorithm based on neural networks. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 706–711. Springer, Heidelberg (2004)
8. Cierniak, R.: New neural network algorithm for image reconstruction from fan-beam projections. Elsevier Science: Neurocomputing 72, 3238–3244 (2009)
9. Cpałka, K.: Cpałka K., 2009. A New Method for Design and Reduction of Neuro-Fuzzy Classification Systems. IEEE Transactions on Neural Networks 20(4), 701–714 (2009)
10. Cpałka, K.: On evolutionary designing and learning of flexible neuro-fuzzy structures for nonlinear classification. Nonlinear Analysis Series A: Theory, Methods and Applications 71(12), e1659–e1672 (2009)
11. Cpałka, K., Rutkowski, L.: A new method for designing and reduction of neuro-fuzzy systems. In: 2006 IEEE International Conference on Fuzzy Systems, pp. 1851–1857 (2006)
12. Cpałka, K., Rutkowski, L.: Flexible Takagi-Sugeno Neuro-Fuzzy Structures for Nonlinear Approximation. WSEAS Transactions on Systems 4(9), 1450–1458 (2005)
13. Fogel, D.B.: Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, 3rd edn. IEEE Press, Piscataway (2006)
14. Gabryel, M., Cpałka, K., Rutkowski, L.: Evolutionary strategies for learning of neuro-fuzzy systems. In: I Workshop on Genetic Fuzzy Systems, Genewa, pp. 119–123 (2005)
15. Gabryel, M., Rutkowski, L.: Evolutionary methods for designing neuro-fuzzy modular systems combined by bagging algorithm. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 398–404. Springer, Heidelberg (2008)
16. Horzyk, A., Tadeusiewicz, R.: Self-Optimizing Neural Networks. In: Yin, F.-L., Wang, J., Guo, C. (eds.) ISNN 2004. LNCS, vol. 3173, pp. 150–155. Springer, Heidelberg (2004)
17. Ogata, K.: Modern Control Engineering. Prentice Hall (2001)
18. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer Academic Publishers (2000)
19. Korytkowski, M., Gabryel, M., Rutkowski, L., Drozda, S.: Evolutionary methods to create interpretable modular system. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 405–413. Springer, Heidelberg (2008)
20. Korytkowski, M., Rutkowski, L., Scherer, R.: On combining backpropagation with boosting. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN), Vancouver, pp. 1274–1277 (2005)
21. Laskowski, Ł.: A novel hybrid-maximum neural network in stereo-matching process. Neural Computing & Applications (2012), doi:10.1007/s00521-012-1202-0.

22. Laskowski, Ł.: Objects auto-selection from stereo-images realised by self-correcting neural network. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part I. LNCS, vol. 7267, pp. 119–125. Springer, Heidelberg (2012)
23. Laskowski, L.: A novel continuous dual mode neural network in stereo-matching process. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010, Part III. LNCS, vol. 6354, pp. 294–297. Springer, Heidelberg (2010)
24. Li, X., Er, M.J., Lim, B.S., et al.: Fuzzy Regression Modeling for Tool Performance Prediction and Degradation Detection. International Journal of Neural Systems 20(5), 405–419 (2010)
25. Nowicki, R., Pokropińska, A.: Information criterions applied to neuro-fuzzy architectures design. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 332–337. Springer, Heidelberg (2004)
26. Nowicki, R., Scherer, R., Rutkowski, L.: A method for learning of hierarchical fuzzy systems. In: Sincak, P., Vascak, J., Kvasnicka, V., Pospichal, J. (eds.) Intelligent Technologies - Theory and Applications, pp. 124–129. IOS Press (2002)
27. Przybył, A.: Doctoral dissertation: Adaptive observer of induction motor using artificial neural networks and evolutionary algorithms. University of Technology, Poznan (2003) (in Polish)
28. Przybył, A., Cpałka, K.: A new method to construct of interpretable models of dynamic systems. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 697–705. Springer, Heidelberg (2012)
29. Przybył, A., Smoląg, J., Kimla, P.: Real-time Ethernet based, distributed control system for the CNC machine. Electrical Review (February 2010) (in Polish)
30. Rutkowska, D., Nowicki, R., Rutkowski, L.: Neuro-fuzzy architectures with various implication operators. In: Sincak, P., et al. (eds.) The State of the Art in Computational Intelligence, pp. 214–219 (2000)
31. Rutkowski, L., Przybył, A., Cpałka, K.: PrzybyłA. and Cpałka K., 2012. Novel Online Speed Profile Generation for Industrial Machine Tool Based on Flexible Neuro-Fuzzy Approximation. IEEE Transactions on Industrial Electronics 59(2), 1238–1247 (2012)
32. Rutkowski, L., Cpałka, K.: Flexible weighted neuro-fuzzy systems. In: Proceedings of the 9th Neural Information Processing, pp. 1857–1861 (2002)
33. Rutkowski, L.: Computational Intelligence. Springer (2007)
34. Rutkowski, L., Cpałka, K.: Flexible neuro-fuzzy systems. IEEE Trans. Neural Networks 14(3), 554–574 (2003)
35. Scherer, R.: Boosting Ensemble of Relational Neuro-fuzzy Systems. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 306–313. Springer, Heidelberg (2006)
36. Scherer, R., Rutkowski, L.: A fuzzy relational system with linguistic antecedent certainty factors. In: 6th International Conference on Neural Networks and Soft Computing, Zakopane, Poland. Book Series Advances In Soft Computing, pp. 563–569 (2003)
37. Scherer, R., Rutkowski, L.: Connectionist fuzzy relational systems, 9th International Conference on Neural Information and Processing; 4th Asia-Pacific Conference on Simulated Evolution and Learning; 1st International Conference on Fuzzy Systems and Knowledge Discovery, Singapore. Computational Intelligence for Modelling and Prediction Book Series: Studie. In: Computational Intelligence, vol. 2, pp.35-47 (2005)

38. Scherer, R., Rutkowski, L.: Neuro-fuzzy relational classifiers. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 376–380. Springer, Heidelberg (2004)
39. Szaleniec, M., Goclon, J., Witko, M., Tadeusiewicz, R.: Application of artificial neural networks and DFT-based parameters for prediction of reaction kinetics of ethylbenzene dehydrogenase. Journal of Computer-Aided Molecular Design 20(3), 145–157 (2006)
40. Zalasiński, M., Cpałka, K.: A new method of on-line signature verification using a flexible fuzzy one-class classifier. Selected Topics in Computer Science Applications, pp. 38–53. EXIT (2011)
41. Zalasiński, M., Cpałka, K.: Novel algorithm for the on-line signature verification. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 362–367. Springer, Heidelberg (2012)
42. Żebrowski, J., Grudziński, K.: Observations and modelling of unusual patterns in human heart rate variability. Acta Physica Polonica B 36(5), 1881–1895 (2005)

Towards Efficient Information Exchange in Fusion Networks

Hela Masri[1], Adel Guitouni[2], and Saoussen Krichen[3]

[1] LARODEC Laboratory, Institut Supérieur de Gestion de Tunis, 2000 Bardo, Tunisia
masri_hela@yahoo.fr
[2] Peter B. Gustavson School of Business, University of Victoria, Victoria (British-Columbia), Canada
Adel.Guitouni@drdc-rddc.gc.ca
[3] Faculty of Law, Economics and Management, University of Jendouba, Avenue de l'UMA, 8189 Jendouba, Tunisia
saoussen.krichen@isg.rnu.tn

Abstract. In this paper, we address information exchange problem in heterogeneous fusion networks (decision networks). A fusion network is a set of connected nodes in which fusion nodes (decision-agents: DAs) consume information produced by other sources nodes (e.g., sensors, other fusion nodes), and information is exchanged across a web of connected nodes. Information value is assessed based on partial utility function. This value, representing the DA's utility, is modeled as a time depending function. Routing in a fusion network is not just about getting data from one point to another. Routing needs to optimize a set of end-to-end goals driven by the application requirements, while considering network resources. We model this problem as a bi-objective optimization problem that maximizes the overall utility of the network and reliability of the generated paths. A multi-objective genetic algorithm (MOGA) is proposed to solve such an NP-hard problem. The empirical results are also presented.

1 Introduction

According to the JDL (1999), Information Fusion is the process of combining data to refine state estimates and predications. The information fusion purpose is to produce information from different sources in order to support the decision-making process. For example, a decision-level identity fusion aim at processing sensors data to obtain identity estimates of target. Identity fusion can be performed on three levels: raw data level, feature level, or decision level [4]. The purpose of a fusion system should be tailored toward supporting a decision-maker or human. Therefore, information fusion system (IFS) is goal driven. It is constrained by the physical and technical constraints. These constraints might include the available sources of information, their quality, environment conditions, processing speed, networking constraints (e.g., available bandwidth,

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 535–546, 2013.

lateness), uncertainties, etc. The goal might be to declare an identity or to assess a given metric (e.g., speed, altitude). Nowadays, information fusion is not necessarily performed by a centralized fusion process. With the advent of social networking, distributed computing and smart sensors, information fusion has become a networked distributed process. Therefore, not only one should examine improvements to fusion itself, but to the information exchange strategies in fusion networks. An efficient information exchange policy needs to be set up to ensure an overall improvement of the knowledge created by the fusion network (e.g., reduction of the overall uncertainties or improvement of the entropy). An information fusion network is characterized by the collaboration of several fusion nodes processing diverse information sources' inputs in order to achieve a global goal. These interactions rely on a communication network supporting the information sharing among the dispersed entities. For example, in the large volume surveillance problem, the execution of any surveillance mission requires a web of heterogeneous communication networks to exchange information and coordinate actions. Therefore, the performance of the execution of a surveillance mission is clearly dependent on the availability of different networks and the information exchange strategies used.

In this paper, we model a network of information fusion nodes and information sources (e.g., sensors) by a decision network. The information fusion nodes are represented as different decision agents (DAs) with different information needs. Other nodes might be considered like relays and transmitters of information. Decision agent can be an information producer or/and information consumer. The information has different value for each DA, which is represented as a time dependent utility function. The network is modeled as a finite set of nodes that can communicate using pre-established connections, and each connection is characterized by several metrics (e.g., predetermined capacity, lead time and cost, reliability). We assume there is a centralized routing coordinator managing all information exchange requests. The problem is therefore to generate optimal information exchange strategy in order to maximize the performance of the information fusion network. In this paper, we study the information exchange in fusion networks from a new perspective, by assuming that a domain specific network might have a purpose behind information sharing, so that the nodes are application aware (fusion aware). Solving the routing problem consists on defining optimal routing plans that optimize the overall information value in the network as well as other quality of service (QoS), while respecting the available resources of the network. We present a mathematical formulation of the this problem as a bi-objective optimization problem by extending the basic formulation of the unsplittable multi-commodity flow problem [3].

This paper is organized as follows. In the next section, we provide an overview of the literature about related routing algorithms. Section 3 presents the problem description. The bi-objective mathematical programming formulation is stated in section 4. Section 5 describes the proposed solution approach. Section 6 provides some experimental results illustrating the efficiency of the proposed method.

2 Literature Review

A large number of paper dealt with information fusion in ad-hoc networks and more precisely in sensor networks [11]. Information fusion arises as a discipline that is concerned with how data gathered by different agents can be processed to increase the relevance of such a mass of data. However, fusion methods do not consider the control of data flows [9]. Given the limited bandwidth of the network, uncontrolled data flows may cause large amounts of conflicting plans that might deteriorates the performance of the whole system. Thus, one crucial problem is how to optimize routing so that relevant information can be fused while respecting the network capacity? Information fusion is considered as high level task that is basically related to application layer requirements, however routing protocols are more related to the network topology and telecommunication infrastructure since it is located in the network layer. Though, routing of information between different fusions nodes and different information sources is proven to influence the quality of the output. Different routing protocols were designed in the literature. These protocols might differ depending on the application and network architecture. An ideal routing algorithm should find an optimum path for packet transmission so as to satisfy some QoS (delay, bandwidth, reliability) [10, 1–3]. However, routing is not generally coupled with the application semantics of the network. Assuming a centralized control, the routing problem is generally modeled as a multicommodity flow (MCFP) problem where multiple pairs of source-destination are managed. A linear version of this problem exists if the demand can be split. However for some applications, it may be required that an information is sent along a single path [3]. Such problem is denoted the unsplittable multicommodity problem, proven to be NP-hard [2]. This problem was investigated only by few researchers in the single objective framework, such as minimizing the cost [2] or minimizing the congestion [3]. In most cases, the problem was solved using approximation algorithms [3]. Alternatively, Barnhart et al. [1] proposed an exact method using branch-and-price-and-cut algorithm. More recently, a metaheuristic approach based on ant colony system method was developed [2]. Only one paper [10] proposed a bi-objective formulation for unsplittable MCFP solved also by ACS. However, none of the above routing algorithms consider the application requirements while routing data.

Recently, some papers proposed some application directed routing algorithms, basically for sensor networks [12, 6, 5]. Wu et al. [12] considered the problem of computing a route for a mobile agent that incrementally fuses the data as it visits the nodes in a distributed sensor network. The order of nodes visited along the route has a significant impact on the quality and cost of fused data. They demonstrate that this problem is NP-complete and proposed a genetic algorithm to compute an approximate solution. Some papers [5, 6] introduced the idea of information-directed routing in which routing problem is formulated as a joint optimization of data transport and information aggregation. Both algorithms [5, 6] are designed for tracking application where a successive message refinement is processed while routing a message. Chu et al. [5] proposed the Constrained anisotropic diffusion routing (CADR) that considers both routing

and data aggregation. The key idea is to introduce an information utility measure to select which sensors to query and to dynamically guide data routing. In this paper we consider a more general context, by assuming that we have a decision network where DAs need to interact in order to share information. The relevance of information is time dependent; therefore we model the value of information as a time dependent utility function assigned by the receiver (a DA). The problem is formulated as a bi-objective non linear problem, that optimizes the overall value of information and the reliability of the paths, subject to structural constraint such as capacity, time window, single path constraints.

3 Problem Description

In this paper, we propose an application driven routing algorithm for heterogeneous networks. We assume that the network layer is aware of the applications' requirements and the purpose behind the information exchange. We address the optimization of information routing in heterogeneous information fusion network, in the bi-objective framework. This problem is about exchanging various messages from a set of sources to different destinations. Each node in the network can be an information provider (source) or/and a DA requiring an information (destination) or simply a relay node. A node can be an automated system like a sensor, a fusion node or a human DA. These nodes are connected across a web of heterogeneous links. An edge is characterized by: A limited capacity c, lead time, a medium type (representing the compatibility of messages that can be sent along each edge) and a reliability. The structure of this network is fixed and the characteristics of the links are deterministic. Heterogeneity means that we might have different types of communication networks (e.g., wireless, radio, Internet). Each network type is represented by a separate set of edges. Given these statements, the network can be modeled as a directed multiple edge graph (N,M) where $N = (v_1, .., v_{|N|})$ is the set of nodes and $M = (e_1, .., e_{|M|})$ is the set of edges. We assume that each DA (or fusion node) has a soft time window to receive the information, and that the same information might be offered by different nodes with different credibility. A finite set of messages is shared by the network nodes. Each message has a compatibility constraint. Furthermore, each decision agent has an a priori value for each information message. Some messages can be transmitted only on a particular type of communication network (one or more). Furthermore, each DA will assign a value to each required information. This value represents the relevance of the information for the DA, and it is modeled as a time dependent decreasing function. It is continuous, and comprised between 0 and 1, according to the time window imposed by the receiver. Figure 1 describes the form of the utility function given the time window $[a, b]$ defined by the DA. The solution to this problem consists on defining how to exchange messages from producers to consumers by assigning a source node to each request, and by generating a transmission path, such that to maximize the overall utility and reliability.

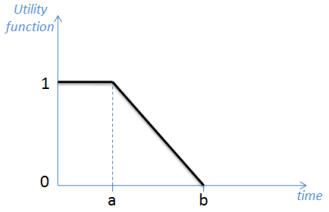


Fig. 1. The utility function of a decision maker

4 Mathematical Programming Formulation

We propose a mathematical programming formulation with two objective functions. This formulation is a mixed integer nonlinear program. It is an extension of the basic mathematical formulation of the MCFP (path formulation [7]). A set of paths relating the possible (source, destination, message) triplets are given as inputs of the model, these paths respect the compatibility constraint according to the type of message. A solution associates to each pair of (destination node, message), an assigned source node and a flow path (i.e succession of edges used to transmit the message). It also generates the schedule of transmission by specifying the transmission start time of a message along each edge of the path. Some of the requests might not be satisfied due to the time window constraint. The mathematical model simultaneously maximizes the total utility and reliability while respecting the source assignment, the capacity, time window, and scheduling constraints.

Notation:

N:	the set of nodes $\{v_1..v_{\|N\|}\}$ of the graph
M:	the set of edges $\{e_1..e_{\|M\|}\}$ of the graph
I:	the set of messages $\{i_1..i_{\|I\|}\}$ to be shared
cr_n:	the credibility the node n
c_m:	the capacity of the edge m
l_m:	the lead time of the edge m
r_m:	the reliability of the edge m
t:	a given discrete time
$size_i$:	the size of the message i
ic_{ni}:	=1 if the node n is a DA requiring the message i =0 otherwise
ip_{ni}:	=1 if the node n is a provider of the message i =0 otherwise

$U_{di}(t)$: the utility function value defined by the node d for receiving the message i at the time t
$[a_{di}, b_{di}]$: a time window for the DA d asking for the message i
x_{di}: =1 if the message i is succefully transmitted to the receiver d.
=0 otherwise
P^{sdi}: the set of all paths that start at s and end at d compatible with the type of message i

$$max \quad Z_1(X) = \sum_s \sum_d \sum_i \sum_k x_{di}\ p_k^{sdi} cr_s U_{di}(T_{di}) \tag{1}$$

$$max\ Z_2(X) = \sum_s \sum_d \sum_i x_{di} \sum_k p_k^{sdi} \prod_{m|\delta_{km}^{sdi}=1} r_m \tag{2}$$

$$\sum_s \sum_k p_k^{sdi} \leq x_{di} \qquad d = 1,..,N \quad i = 1..I \tag{3}$$

$$p_k^{sdi} \leq ip_{si} \qquad d = 1,..,N \quad i = 1..I \quad s = 1,..,N \quad k \in P^{sdi} \tag{4}$$

$$p_k^{sdi} \leq ic_{di} \qquad d = 1,..,N \quad i = 1..I \quad s = 1,..,N \quad k \in P^{sdi} \tag{5}$$

$$ET_{sdik}^m \leq ST_{sdik}^{m'} \qquad s = 1...,N \ \ d = 1,..,N \ \ i = 1..I \quad k \in P^{sdi} \ \ m = m_{sdik}^{(j)} \ \ m' = m_{sdik}^{(j+1)} \ \ j = 1..(l_{sdik} - 1) \tag{6}$$

$$T_{di} = \sum_s \sum_k p_k^{sdi} ET_{sdik}^m \qquad d = 1,..,N \quad i = 1..I \quad m = m_{sdik}^{(l_{sdik})} \tag{7}$$

$$x_{di}\ T_{di} \leq b_{di} \qquad d = 1,..,N \quad i = 1..I \tag{8}$$

$$ET_{sdik}^m = p_k^{sdi}(ST_{sdik}^m + l_m + \tfrac{size_i}{c_m}) \qquad m = 1..M \ \ d = 1,..,N \ \ i = 1..I \ \ s = 1,..,N \tag{9}$$

$$(ET_{sdik}^m - t)(t - ST_{sdik}^m) \leq \sum_k p_k^{sdi}\ \delta_{km}^{sdi}\ y_{sdim}^t\ G\ (ET_{sdik}^m - t)(t - ST_{sdik}^m) \qquad t = 1..T \ \ m = 1..M \ \ d = 1,..,N \ \ i = 1..I \ \ s = 1,..,N \tag{10}$$

$$\sum_s \sum_d \sum_i y_{sdim}^t \leq 1 \qquad m = 1..M \quad t = 1..T \tag{11}$$

$$x_{di} \in \{0,1\}, \quad p_k^{sdi} \in \{0,1\}, \quad ST_{sdik}^m \geq 0, \quad y_{sdim}^t \in \{0,1\} \qquad m = 1,..,M \ \ d = 1,..,N \ \ s = 1,..,N \ \ i = 1..I \ \ k \in P^{sdi} \tag{12}$$

p_k^{sdi}: =1 if the path k is used to satisfy the flow (s, d, i)
δ_{km}^{sdi}: =1 if the edge m exists on the path k used to satisfy the flow (s, d, i)
=0 otherwise
G: a large value
T_{di}: the total transmission time of the generated path to satisfy (d, i)
l_{sdik}: the total number of edges composing the path of the flow (s, d, i).
$m_{sdik}^{(j)}$: the j^th edge in the path k used for the flow (s, d, i)
ST_{sdik}^m: the transmission's start time of the message i along the edge m in order to satisfy the flow (s, d, i)
y_{sdim}^t: =1 if the edge m is used by the flow (s, d, i) at instant t
=0 otherwise

The mathematical formulation is stated (1)-(11). The two considered objective functions are to:

- Maximize the utility of the whole network which is equal to the sum of the utility functions of each flow weighted by the credibility of the assigned source node. The utility function is dependent of the arrival time of the required message.
- Maximize the overall reliability of the generated paths of the satisfied flows.

Constraint (3) verifies that each satisfied request, described by a pair of destination-message (d, i), is assigned exactly to one path linking its assigned source s and its destination d. Constraints (4)-(5) represent the source assignment constraints, it ensures that each a pair of (destination node, message) (d, i) should be assigned to one source node s, such that s is one of the possible producers of the required message and d a requester. Constraint (6) represents the precedence constraint of the transmission start time between the edges composing a path. So that, the transmission end time on the j^{th} edge of a path should be lower than the transmission start time on the next $(j+1)^{th}$ edge. Constraints (7) and (8) ensure that the arrival time T_{di} of a flow should respect the time window upper bound b_{di}. Constraint (10) states that for a given edge m and a flow (s, d, i), y^t_{sdim} should be equal to one for all instant t comprised between the transmission start time and the arrival time along this edge. This constraint means that if an edge m belongs to an activated path k (i.e $p^{sdi}_k = 1$ and $\delta^{sdi}_{km} = 1$), and for a given time t, if $(ET^m_{sdik} - t) \geq 0$ and $(t - ST^m_{sdik}) \geq 0$, then y^t_{sdim} is equal to 1, and 0 otherwise. Finally, the capacity constraint (11) means that at a given time, only one flow is passing through an edge m.

5 Solution Approach

The combinatorial structure of the proposed model makes generation of the solution difficult and time consuming. In addition, the complexity of the problem is NP-hard due to the single path constraint. These reasons justify the computational impracticality of exact algorithms for solving this problem. Therefore, we propose to solve it using a metaheuristic method, based on a multiobjective genetic algorithm (MOGA). Owing to the distinctive features such as domain independence, non-linearity, robustness and parallel nature, MOGAs have been proved to be an effective approach for solving optimization problems. The successful application of genetic algorithms for solving similar routing problems motivated the choice of the metaheuristic. Starting with a random population, the individuals evolve to new solutions that approximate better the pareto front. This algorithm maintains an evolutionary population P and an external non dominated solution set P_{ND}. The basic outline of the algorithm is described, and each procedure is briefly explained.

Genetic algorithm

i=0
Generate first population P_0 of size N
Define the optimal schedule for each solution
repeat
  $i = i + 1$
  Crossover: generate the set R_i of offspring of size N
  Define the optimal schedule for each solution in R_i
  Update the non dominated set
  $P_i = P_{i-1} \cup R_i$
  Evaluate and select best N solution from P_i
until (stopping criteria)

5.1 Chromosome Representation

Better efficiency of GA-based search could be achieved by well defining the chromosome representation and its related operators so as to generate feasible solutions and avoid repair mechanism. Figure 2 depicts the structure of a chromosome. The adopted representation is a variable length with bounded solution size; since the size of a substring vary with the number of the edges figuring in the path. A chromosome is coded as a vector of R substrings where R is the number of requests. Each substring is composed of two parts::

- A bit describing if this request is satisfied or not.
- Two vectors describing the combination of path assignment and transmission scheduling decisions, expressed by the list of indexes of the used edges composing the path and the sequence of the transmission start time on these edges.

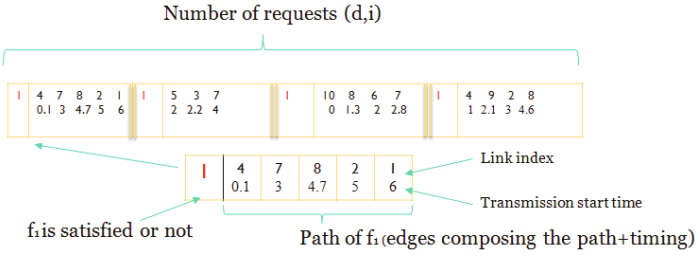


Fig. 2. Chromosome representation

5.2 Generating the First Population

In order to find the initial population, a greedy algorithm based on nearest neighbour method is used, weights are randomly generated in order to aggregate the two objective functions. After setting the size of the population, for each solution, we generate separately a feasible path for each request. The paths are dynamically built such that a partial solution is constructed sequentially following a probabilistic model. The choice of a neighbour depends on the reliability and utility of an edge. These two measures are aggregated according to random weights. A random wheel selection is applied to choose the best edge (i.e the next node to be explore in the path). We use a reverse path construction strategy so that we start from the destinations until reaching an adequate source. Then, for each generated chromosome of the first population, a greedy scheduling algorithm is applied to find the transmission start time of the messages on each edge.

5.3 Crossover

The proposed algorithm applies the random key method for the crossover, the offsprings are produced from two parent solutions following these steps. Two individuals are randomly selected from the current population to act as parents. For each substring a random number between [0,1] is generated. If the generated number is smaller than a threshold value, the substring of the first parent is copied into the offptring chromosome. Otherwise, the gene of the second parent is used. Only the paths are involved in the crossover. Later on, the schedule will be modified according to the result of the crossover.

5.4 Generating the Schedule

After generating N new offsprings, a schedule is defined for each new solution. The schedule is based on a greedy algorithm. Given a feasible solution where the paths are already set, the algorithm computes for each request the arrival time of the corresponding message. Then, it accordingly ranks the flows to be satisfied first, by subserving the requests that have the most restrictive time windows. After defining the ranking, the requests are satisfied in order of their rank. Therefore, the edges are considered as resources to be shared. A list recording the intervals of occupancy of each edge is managed. If the arrival time at a destination exceeds the time window, the corresponding request is considered as not satisfied.

5.5 Selection Procedure

After generating the offsprings, a non domination sorting is applied in order to evaluate the obtained solutions. The N best solutions are selected to be the population of the next iteration. This procedure ranks the solutions of the current population based on non-domination. All the non-dominated individuals in the current population are placed at the top of a list and assigned rank 1. These solutions are removed from the remaining population and the next set of non-dominated solutions is identified and assigned rank 2. The process is repeated until the entire population is ranked. The top N individuals in the list are then selected for the next generation.

6 Experimental Results

A computational analysis is performed in a test bed, in order to evaluate the proposed algorithm. As a simulation environment, we used INFORM-Lab Simulation testbed [8]. This environment enables to execute different algorithms for distributed information fusion and dynamic resource management. It contains different surveillance vignettes with different scenarios. The instances are generated based on a real network: the US National Science Foundation (NSF) [6, 2], containing 14 nodes and 42 directed arcs. The simulation parameter to vary is the number of messages to be shared. For each message we generate a random number of requests and offers. For each request we assume that the time window is assigned randomly. Also for each offer, we generate randomly the size of the information as well as the accuracy of the information (random but positively correlated to the size). A total number of 10 instances are generated, by varying the number of messages between 10 and 100. Table 1 describes the configuration used to generate each instance. We assume that each parameter is uniformly distributed in the defined interval. We choose the maximum number of iterations as

Table 1. Experiments design

Parameter	Interval
Number of requests	[1..2]
Number of offers	[2..3]
Message size	[1..10]
Accuracy	$[0.5..1].(1 - \frac{1}{size})$
Time window	$a \in [0, 50]$
	$b \in [50, 300]$

the termination criteria, set to 1000 iterations. Also if there is no improvement after 30 iterations, the algorithm stops (i.e if all the generated solutions over the last 30 iterations are dominated). The crossover probability is set to 0.5, and the population size is equal to 50. Each Instance is solved with 10 independent runs. We report in table 1 the average of the following measures: number of messages to be shared $NbMsg$, number of offers $NbOff$ and requests $NbReq$, the CPU time, the size of the non dominated set $|P_{ND}|$, the best utility U and reliability R of all the generated non dominated solutions, the CPU time for generating the first population FP, and the number of iterations $NbIter$. By analyzing these results, we can notice that:

- The CPU time for generating the first population represents approximately 10% of the total CPU time required to solve each instance. This fact is due to the constructive nature of the procedure used in the first population.
- The size of the non-dominated set increases with the instance's size. Generally, when the number of messages increases, the number of possible paths' combinations grows. Hence, the number of diversified potentially efficient solutions becomes larger. For example, if the number of messages is equal to

Table 2. Experimental results

	Problem description		**Results**						
Problem	Nb Msg	Nb Off	Nb Req	CPU(s)	$\|P_{ND}\|$	U	R	FP CPU(s)	$NbIter$
1	10	24	11	0.74	9	7.95	7.71	0.37	67.8
2	20	53	28	2.57	16.4	18.54	20.38	0.49	183.6
3	30	74	32	4.62	19	29.05	32.03	0.79	179.6
4	40	115	52	5.27	12.2	38.6	44.12	1	145.4
5	50	130	86	6.10	12.2	47.61	54.25	0.94	167.2
6	60	134	96	10.10	21	45.68	67.01	1.26	197.2
7	70	174	102	12.89	17.6	51.45	72.29	1.71	166.2
8	80	196	135	17.35	19.4	59.37	85.30	1.96	196.2
9	90	238	142	18.05	19.4	60.62	91.78	2.09	210.6
10	100	264	153	20.02	19.8	68.55	106.95	2.29	169

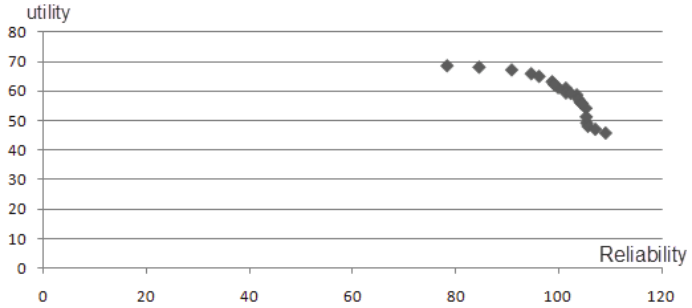


Fig. 3. The non dominated set of problem 10

10, the average size of the non dominated set is 9. However, if 100 messages have to be routed the algorithm generates 20 potentially efficient solutions.

– The algorithm shows a fast convergence. In fact, when solving the instances enumerated in table iv, the algorithm does not elapsed the 1000 iterations prefixed in our setting, it rather finds that no improvements were recorded after an average of 200 iterations.
– Figure 3 displays the non dominated solutions generated by the genetic algorithm for the problem 10 of table 2, 21 potentially efficient solutions were found. We notice that the generated solutions are well scattered. Furthermore, we notice that the Pareto front is non convex. This can be explained by the fact that the mathematical formulation (1)-(10) is not linear.

7 Conclusion

In this paper, we introduced the information fusion networks (or multi-agents decision network). Then, we focused on information exchange optimization problem in these networks. In fact, we believe that efficient information exchange strategy would contribute to improving the efficiency and performance of distributed information fusion networks (ex., sensors network, large volume surveillance, etc).

We proposed a joint information routing and scheduling problem to maximize the overall information value in distributed information fusion networks. We proposed a bi-objective mathematical formulation modeling a preemptive routing. Two main objectives have to be optimized: maximize the overall information value and the reliability of the paths, under some constraints. Compared to the existing literature, the proposed approach seems to be very promising since we are handling the routing problem from a different perspective by including the value of information as a principal metric. To address the complexity of the problem (NP-hard), we proposed an adaptation of a genetic algorithm. The algorithm efficiency was validated by computational experiments on different instances.

References

1. Barnhart, C., Hane, C.A., Vance, P.H.: Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. Operations Research 48(2), 318–326 (2000)
2. Barnhart, C., Hane, C.A., Vance, P.H.: An ant colony optimization metaheuristic for single-path multicommodity network flow problems. Journal of the Operational Research Society 61(9), 1340–1355 (2010)
3. Bley, A.: Approximability of unsplittable shortest path routing problems. Networks 54(1), 23–46 (2009)
4. Bosse, E., Guitouni, A., Valin, P.: An Essay to Characterise Information Fusion Systems. In: 9th International Conference on In Information Fusion, pp. 1–7 (2006)
5. Chu, M., Haussecker, H., Zhao, F.: Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. Int. J. High-Performance Comput. Appl. 16(3), 293–313 (2002)
6. Liu, J., Feng, Z., Petrovic, D.: Information-directed routing in ad hoc sensor networks. IEEE Journal on Selected Areas in Communications 23(4), 851–861 (2005)
7. Holmberg, K., Yuan, D.: A Multicommodity Network-Flow Problem with Side Constraints on Paths Solved by Column Generation. INFORMS Journal on Computing 57(1), 42–57 (2003)
8. MacDonald, Dettwiler and Associates Ltd., Inform Lab Wiki, `https://xwiki.mdacorporation.com/InformLab/www/wiki/view/Main/WebHome`
9. Manyika, J., Durrant-Whyte, H.: Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach. Prentice Hall PTR (1995)
10. Masri, H., Krichen, S., Guitouni, A.: An ant colony optimization metaheuristic for solving bi-objective multi-sources multicommodity communication flow problem. In: 4th Joint IFIP Proceeding of Wireless and Mobile Networking Conference (WMNC), pp. 1–8 (2011)
11. Nakamura, E.F., Loureiro, A.A.F., Frery, A.C.: Information fusion for wireless sensor networks: Methods, models, and classifications. ACM Comput. Surv. 39(3), 9 (2007)
12. Wu, Q., Rao, N.S.V., Barhen, J., Iyengar, S.S., Vaishanavi, V.K., Qi, H., Chakrabarty, K.: On computing mobile agent routes for data fusion in distributed sensor networks. IEEE Transactions on Knowledge and Data Engineering 16(6), 740–753 (2004)

Application of Particle Swarm Optimization Algorithm to Dynamic Vehicle Routing Problem

Michał Okulewicz and Jacek Mańdziuk

Warsaw University of Technology,
Faculty of Mathematics and Information Science,
Koszykowa 75, 00-662 Warsaw, Poland
{M.Okulewicz,J.Mandziuk}@mini.pw.edu.pl

Abstract. In this paper we apply Particle Swarm Optimization (PSO) algorithm to Dynamic Vehicle Routing Problem (DVRP) for solving the client assignment problem and optimizing the routes. Our approach to solving the DVRP consists of two stages. First we apply an instance of PSO to finding clients' requests that should be satisfied by the same vehicle (thus solving the Bin Packing Problem (BPP) part of the DVRP task) and then we apply several other instances of PSO, one for each of the vehicles assigned to particular requests (thus solving the Travelling Salesman Problem (TSP)). Proposed algorithm found better solutions then other tested PSO approaches in 6 out of 21 benchmarks and better average solutions in 5 of them.

Keywords: Particle Swarm Optimization, Dynamic Vehicle Routing Problem, Dynamic Optimization.

1 Introduction

The goal of Vehicle Routing Problem (VRP) is to find the shortest routes for n homogeneous vehicles delivering cargo for m clients. Every client has its own demand of cargo. The capacity of each vehicle is limited and the cargo could be loaded on the vehicle on one of the k depots.

VRP could be regarded as a composition of two NPC problems: the Bin Packing Problem (optimizing the number of used vehicles by assigning client's requests to the vehicles) and the Travelling Salesman Problem (optimizing the route for each of the vehicles). In this study we are focused on the dynamic version of the problem (DVRP) where clients' requests (load demands and required destinations) are not fully available beforehand and are partly defined during the working day (the so-called Vehicle Routing Problem with Dynamic Requests [11]). The goal of DVRP is to find the shortest routes within the time bounds of the working day (i.e. the opening hours of the depots).

In recent years, biologically inspired computational intelligence algorithms have grown a lot of popularity. Examples of such algorithms, based on the idea of swarm intelligence, are PSO and bird flocking algorithm. Those algorithms have been used in the real world applications in the area of computer graphics and animation [16], detection of outliers [3] or document clustering [6].

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 547–558, 2013.

```
PSO() {
  swarm.initializeRandomlyParticlesLocationAndVelocity();
  for i from 1 to maxIterations {
    for each particle in swarm {
      particle.updateVelocity();
      particle.updateLocation();
    }
  }
}

Particle {
  updateVelocity() {
    for (i from 1 to dimensions) {
      this.v[i] =
        random.uniform(0,g)*(this.neighbours.best[i] - this.x[i]) +
        random.uniform(0,l)*(this.best[i] - this.x[i]) +
        random.uniform(0,r)*(this.x[i] - this.neighbours.random().x[i]) +
        a * this.v[i] +
        y * random.normal(0,1);
    }
  }

  updateLocation() {
    for (i from 1 to dimensions) {
      this.x[i] = this.x[i] + this.v[i];
    }
    if (f(this.best) > f(this.x)) {
      this.best = this.x;
    }
  }
}
```

Fig. 1. Particle Swarm Optimization in pseudo-code

PSO was proven to be a suitable algorithm for solving dynamic problems [4] and was applied with success to the DVRP[10,9] with the combination of 2-Opt algorithm for solving the TSP part of the problem. In this work, we employ a different approach, by using PSO algorithm for both phases and furthermore by applying a different problem encoding in the first phase.

One of our goals was to check the possibility of using a continuous encoding of the DVRP problem, thus enabling the usage of the native (continuous) version of the PSO algorithm. The other goal was to check how good results might be achieved, without using the approximation algorithm and relying only on the PSO in both parts of the problem.

The remainder of the paper is organized as follows: First, in section 2 we briefly summarize the PSO algorithm. In section 3 the DVRP is defined in a formal way. Application of the PSO algorithm and the experimental setup and results are presented in sections 4 and 5, respectively. The last section summarizes the experimental findings and concludes the paper.

2 Particle Swarm Optimization Algorithm

PSO algorithm is an iterative optimization method proposed in 1995 by Kennedy and Eberhart [8] and further studied and developed by many other researchers, e.g., [18], [17], [5]. In short, PSO implements the idea of swarm intelligence to solving hard optimization tasks.

In the PSO algorithm, the optimization is performed by the set of particles which are communicating with each other (see Fig. 1). Each particle has its location and velocity. In every step t a location of particle i, x_t^i is updated based on particle's velocity v_t^i:

$$x_{t+1}^i = x_t^i + v_t^i. \tag{1}$$

In our implementation of PSO (based on [1] and [18]) particle's i velocity v_t^i is updated according to the following rule:

$$\begin{aligned} v_{t+1}^i =& u_{U[0;g]}^{(1)}(x_{best}^{neighbours_i} - x_t^i) + u_{U[0;l]}^{(2)}(x_{best}^i - x_t^i) + \\ & u_{U[0;r]}^{(3)}(x_t^i - x_t^{random_i}) + av_t^i + fy_{N(0;1)}, \end{aligned} \tag{2}$$

where

- $x_{best}^{neighbours_i}$ represents the best location in terms of optimization, found hitherto by the neighbourhood of the ith particle,
- x_{best}^i represents the best location in terms of optimization, found hitherto by the particle i,
- $x_t^{random_i}$ is a location of a randomly chosen particle from the neighbours of the ith particle in iteration t,
- g is a neighbourhood attraction factor,
- l is a local attraction factor,
- r is a repulse factor,
- a is an inertia coefficient,
- a is a fluctuation coefficient,
- $u^{(1)}$, $u^{(2)}$, $u^{(3)}$ are random vectors with uniform distribution from the intervals $[0, g]$, $[0, l]$ and $[0, r]$, respectively,
- y is a random vector with coordinates from standard normal distribution.

Such implementation allows to run algorithm as a classic PSO or as a Repulsive PSO (RPSO). In the classic variant $r = 0$ and $g \neq 0$. In RPSO $r \neq 0$ and $g = 0$ (the attraction of the $x_{best}^{neighbours_i}$ point is changed into repulsion from randomly chosen particle allowing for a greater disperse of a swarm).

3 Problem Definition

In the Dynamic Vehicle Routing Problem one considers a fleet V of n vehicles and a series C of m clients (requests) to be served.

All vehicles $v_i, i = 1, \ldots, n$ are identical and have the same *capacity* $\in \mathbb{R}$ and the same *speed*[1] $\in \mathbb{R}$. Each vehicle is loaded in one of the k depots[2].

Each depot $d_j, j = 1, \ldots, k$ has a certain *location* $\in \mathbb{R}^2$ and working hours $(start, end)$, where $0 \leq start < end$.

[1] In all benchmarks used in this paper *speed* is defined as one distance unit per one time unit.

[2] In all benchmarks used in this paper $k = 1$.

Each client $c_l, l = 1+k, \ldots, m+k$ has a given $location \in \mathbb{R}^2$, $time \in \mathbb{R}$, which is a period of time when the request becomes available ($\min_{j \in 1,\ldots,k} d_j.start \leq time \leq \max_{j \in 1,\ldots,k} d_j end$), $unloadTime \in \mathbb{R}$, which is the time required to unload the cargo, and the $requestSize \in \mathbb{R}$ - size of the request ($requestSize < capacity$).

A travel distance $\rho(location_i, location_j)$ is an Euclidean distance between $location_i$ and $location_j$ on the $\mathbb{R}^2$ plane, $i, j = 1, \ldots, m+k$.

As previously stated, the goal is to serve the clients (requests), according to their defined constraints, with minimal total cost (travel distance). Because of the assignment of the requests to the vehicles each vehicle will gain the following properties:

- $PlannedClients_t \subset C$, a set of clients planned to be visited (at a given moment t),
- $AssignedClients_t \subset C$, an ordered set of clients (at a given moment t), which have already been visited or will be visited by the vehicle (the order of the set is defined by the order of visits),
- $distance_i \in \mathbb{R}$, a total travel distance of the vehicle,
- $\widehat{distance}_i \in \mathbb{R}$, an estimation of the total travel distance of the vehicle (based on the clients in the $PlannedClients_t$ set but not on the route through them),
- $depot \in \{1, 2, \ldots, k\}$, the index of the depot to which the vehicle will return at the end of working day.

The sets of the $PlannedClients$ and $AssignedClients$ have the following properties:

$$(\forall_{t \in \mathbb{R}})(\forall_{v \in V})(\forall_{c \in C}) \quad c \in v.PlannedClients_t \Leftrightarrow c \notin v.AssignedClients_t \tag{3}$$

$$\begin{aligned} &(\forall_{t \in \mathbb{R}})(\forall_{v,v\prime \in V})(\forall_{c \in v.PlannedClients_t \cup v.AssignedClients_t}) \\ &c \in v\prime.PlannedClients_t \cup v\prime.AssignedClients_t \Rightarrow v = v\prime \end{aligned} \tag{4}$$

The formulae state that each client could be associated with only one vehicle (eq. 4) and only using one type of association (eq. 3).

4 A 2-Phase Particle Swarm Optimization in Dynamic Vehicle Routing Problem

In the rest of the paper we will use the following definitions:

Available vehicles - a set of vehicles to which the requests may still be assigned (added) without breaking the time bounds of the problem.

Operating area - an area enclosed by the polygon spanned by the locations of planned and assigned clients.

Cut-off time - a time of the day (within the working hours) after which requests (defined in benchmark sets) are postponed to the next working day (they are available at the next day's start time).

4.1 The Algorithm

The problem will be solved by dividing the working day into pairwise equal time slices and solving partial problems with the use of the best known solution previously found.

The number of time slices n_{ts} equals 25 and cut-off time T_{co} equals 0.5 as proposed and tested by Montemanni et al.[15]. We have stuck to the above-mentioned selection in order to allow a direct comparison of our results with the ones obtained previously for this problem [10,9,15,7]. We also measure the dynamism of the problem (*dod*) used by Khouadjia et al. [10], defined as follows:

$$dod = \frac{\text{Number of requests unknown at the beginning of the working day}}{\text{Number of all requests}} \tag{5}$$

Because of the nature of the problem (which is essentially a combination of two different NPC problems) the optimization algorithm has been divided into two separate phases (stages) in each time slice. In the first phase we assign clients to the vehicles and in the second one optimize a route, separately for each of the vehicles.

The optimization is performed for the client requests known at the moment of the algorithm's start (n_{ts} times a day), i.e. the information about new requests is not updated during the algorithm's run (in a given time step).

The general pseudocode of the algorithm is presented on Fig. 2.

4.2 Client Assignment Encoding

The arguments of the fitness function for the first stage are the locations of the centres of the operating areas for the vehicles and the result of the function is a sum of the estimated routes' lengths and optionally an additional penalty for the estimated late return to the depot (i.e. after its closing).

If $x = [x_1, x_2, \ldots, x_{2n}]$ is the argument of the fitness function then the center of the operation area of $vehicle_j$ is defined as follows: $center_j.x = x_{2j-1}$ and $center_j.y = x_{2j}$. The closest (in terms of Euclidean distance between the client location and the vehicle center of the operating area) clients are added to the list of the *planned clients* of the closest **available vehicle**.

The value of the vehicles' routes length estimation equals to the sum of the estimated route length for each of the vehicles $\sum_{v \in V} v.\widehat{distance}$. The vehicle's route length estimation is expressed as a sum of three factors: $assignedRoute$

```
time = depot.start;
while (time < depot.end)
{
  foreach (v in Vehicles)
    v.Available = true;
  while (|UnassignedClients| > 0 and |AvailableVehicles| > 0)
  AssignClientsToAvailableVehicles();
  foreach (v in Vehicles)
  {
    v.OptimizeRoute();
    if (|v.LateClients| > 0)
      v.Available = false;
    v.RemoveLateClientsFromTheRoute(); //mark late clients as unassgined
    v.CommitVehicles();
  }
  time += (depot.end - depot.start) / n_ts;
}
```

Fig. 2. The pseudocode of the algorithm for applying PSO to the DVRP

(length of route through the *assigned clients*), $\widehat{plannedRoute}$ (estimated length of the route through the *planned clients*) and $\widehat{clusterCost}$ (estimated distance from depot to the planned clients multiplied by the doubled number of necessary returns to the depot for reloading the vehicle).

$$v.\widehat{distance} = v.assignedRoute + v.\widehat{plannedRoute} + v.\widehat{clusterCost} \tag{6}$$

$$v.assignedRoute = \sum_{i=2}^{|v.AssignedClients|} \rho\left(v.AssignedClients_{i-1}.location, v.AssignedClients_i.location\right) \tag{7}$$

$$v.\widehat{plannedRoute} = \sum_{i=1}^{|v.PlannedClients|} \rho(v.PlannedClients_i.location, v.center) \tag{8}$$

$$v.\widehat{clusterCost} = \left\lceil \frac{\sum_{c \in v.AssignedClients \cup v.PlannedClients} c.requestSize}{v.capacity} \right\rceil * 2\rho(v.center, v.depot.location) \tag{9}$$

The route through the *planned clients* is estimated by the sum of the distance from the vehicle's operating area center to each of the *planned clients*. And the

distance from the depot to the clients is estimated by the distance from vehicle's operating area center to depot location.

4.3 Managing the Vehicles

Vehicle Route Encoding. The argument of the fitness function for the second stage defines an order of the *planned clients* and the result of the function is the length of the route.

If $x = [x_1, x_2, \ldots, x_{2n}]$ is the argument of the function then the rank of the vehicle's j^{th} *planned client* is equal to the rank of the x_j coordinate in the set $\{x_1, x_2, \ldots, x_{2n}\}$ (if $x_k = x_l \wedge k \neq l$ then the order of the k^{th} and l^{th} clients is defined by the order of l and k).

Committing the Vehicles. After the second phase of the optimization the algorithm returns proposed routes over the *planned clients* set for each of the vehicles. The vehicles with small time reserve[3] are committed to their subsequent *planned clients*.

The set of the clients which are to be moved from the *plannedClients* to the *assigned clients* of the vehicle v is defined as follows:

$$\{client : client \in v.PlannedClients \wedge client.arrivalTime \leq \min_{c \in v.PlannedClients} (c.arrivalTime > nextTimeStep)\}$$

where $arrvialTime$ is the time when the client will be visited by the vehicle v on the planned route.

Keeping Solution within the Time Bounds. Keeping solution within the time bounds is achieved by adding a simple penalty term $pf(vehicles)$ to the vehicles' routes length estimation function. If despite the pf function, the time bounds are exceeded, the late client requests are assigned to the closest **available vehicle**, thus obtaining the feasible solution.

The penalty function is defined as follows:

$$pf(vehicles) = \sum_{v \in vehicles} \begin{cases} (v.\widehat{time} - v.depot.end)^2, & v.\widehat{time} > v.depot.end \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $v.depot$ is the depot to which the vehicle returns at the end of the working day and $v.\widehat{time}$ is vehicle estimated travel time. $v.\widehat{time}$ is equal to the value of the route length evaluation function for the vehicle[4] plus time required to unload the cargo.

[3] Small time reserve was experimentally set as arriving at the depot later then 3 time steps before the depot closing time

[4] The time can by unified with the distance since the vehicle speed is defined as one distance unit per one time unit.

Table 1. Presentation of the best and $\overline{x}$ (average) results with the S (standard deviation of the sample), ratio to the best known VRP solution accomplished by the proposed method (denoted by 2-phase PSO).

Benchmark[13]	2-phase PSO				
	Best	$\overline{x}$	S	$\frac{\text{Best}}{\text{VRPBest}}$ [2]	dod
c50D.vrp	582.88	675.14	40.62	1.11	0.46
c75D.vrp	912.23	1015.16	51.41	1.09	0.52
c100D.vrp	996.40	1149.48	79.78	1.21	0.59
c100bD.vrp	828.94	850.68	23.70	1.01	0.59
c120D.vrp	1087.04	1212.38	106.80	1.04	0.42
c150D.vrp	1173.94	1336.84	62.72	1.14	0.47
c199D.vrp	1446.93	1578.99	71.24	1.12	0.47
f71D.vrp	315.00	356.75	23.70	1.30	0.59
f134D.vrp	12813.14	13491.60	319.41	1.10	0.57
tai75aD.vrp	1871.06	2142.07	162.75	1.16	0.56
tai75bD.vrp	1460.95	1568.21	69.80	1.09	0.56
tai75cD.vrp	1500.23	1811.08	152.80	1.16	0.56
tai75dD.vrp	1462.82	1586.28	95.95	1.07	0.37
tai100aD.vrp	2317.76	2707.61	202.75	1.13	0.56
tai100bD.vrp	2187.86	2510.60	167.79	1.13	0.56
tai100cD.vrp	1564.25	1672.33	68.78	1.11	0.46
tai100dD.vrp	1859.70	2220.01	147.32	1.18	0.49
tai150aD.vrp	3638.75	4151.31	314.32	1.19	0.46
tai150bD.vrp	3107.95	3302.94	126.36	1.14	0.49
tai150cD.vrp	2781.02	2952.88	68.80	1.18	0.49
tai150dD.vrp	3048.24	3478.49	225.68	1.15	0.56
Sum/Avarage	46957.09	51770.84	2582.48	1.13	0.51

Passing Down Historic Knowledge. As the center of the search space is biased in the search for the global optimum [12,19], the center of the search space for the particular time slice is chosen based on the best known solution from the previous time slice while missing arguments (parameters defining the order of requests that were not present before) are generated at random with the underlying condition that at least one particle's initial location is equal to the best solution previously found.

5 Tests and Results

Benchmark Sets. The algorithm was tested on the same benchmark sets as the ones used in [10,9,15,7]. The best achieved value, the mean value, the standard deviation, the relation to the best known static solution and the value of *dod* for each of the tested problems are presented in Table 1. Table 2 shows the best achieved values of the algorithm compared to other PSO approaches [10,9] for solving the DVRP.

Table 2. Comparison of the best and average results accomplished by various approaches and the best known solutions for the static version of considered benchmark problems. The best results in each case are bolded. Our method is presented as 2-phase PSO.

Benchmark[13]	Algorithm						VRP[2]
	2-phase PSO		MAPSO[9]		DAPSO[10]		
	Best	Avarage	Best	Avarage	Best	Avarage	
c50	582.88	675.14	**571.34**	**610.67**	575.89	632.38	524.61
c75	**912.23**	1015.16	931.59	**965.53**	970.45	1031.76	835.26
c100	996.40	1149.48	**953.79**	**973.01**	988.27	1051.50	826.14
c100b	**828.94**	**850.68**	866.42	882.39	924.32	964.47	819.56
c120	**1087.04**	**1212.38**	1223.49	1295.79	1276.88	1457.22	1042.11
c150	**1173.94**	**1336.84**	1300.43	1357.71	1371.08	1470.95	1028.42
c199	**1446.93**	**1578.99**	1595.97	1646.37	1640.40	1818.55	1291.45
f71	315.00	356.75	287.51	**296.76**	**279.52**	312.35	241.97
f134	**12813.14**	**13491.60**	15150.50	16193.00	15875.00	16645.89	11629.60
tai75a	1871.06	2142.07	**1794.38**	**1849.37**	1816.07	1935.28	1618.36
tai75b	1460.95	1568.21	**1396.42**	**1426.67**	1447.39	1484.73	1344.64
tai75c	1500.23	1811.08	1483.10	**1518.65**	**1481.35**	1664.40	1291.01
tai75d	1462.82	1586.28	**1391.99**	**1413.83**	1414.28	1493.47	1365.42
tai100a	2317.76	2707.61	**2178.86**	**2214.61**	2249.84	2370.58	2047.90
tai100b	2187.86	2510.60	**2140.57**	**2218.58**	2238.42	2385.54	1940.61
tai100c	1564.25	1672.33	**1490.40**	**1550.63**	1532.56	1627.32	1407.44
tai100d	1859.70	2220.01	**1838.75**	**1928.69**	1955.06	2123.90	1581.25
tai150a	3638.75	4151.31	**3273.24**	**3389.97**	3400.33	3612.79	3055.23
tai150b	3107.95	3302.94	**2861.91**	**2956.84**	3013.99	3232.11	2727.99
tai150c	2781.02	2952.88	**2512.01**	**2671.35**	2714.34	2875.93	2362.79
tai150d	3048.24	3478.49	**2861.46**	**2989.24**	3025.43	3347.60	2655.67
Sum	**46957.09**	51770.84	48104.13	**50349.66**	50190.87	53538.72	41637.43

Tests Configuration. For all tests the system was run with the following parameters, suggested in [5,18]: $g = 0.6$, $l = 2.2$, $r = 0.0$, $a = 0.63$, $f = 0.0$ in eq. (2). A random star topology with probability $P(X \text{ is a neighbour of } Y) = 0.7$ was applied[5].

The algorithm was run 150 times for each of the benchmark problems. In each case, the number of particles was equal to 40, the number of iterations for each time slice was restricted to 500 (in each of the two phases of the algorithm). Thus the total number of function evaluations for the first phase was equal to 500 000 and the number of function evaluations for the second phase was about 4 000 000 (depending on the number of vehicle routes).

Choice of the Parameters. Parameter $P(X \text{ is a neighbour of } Y) = 0.7$ was empirically chosen from the set of $\{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ based on a small

[5] Note, that the fact that X is a neighbour of Y does not mean that Y is a neighbour of X.

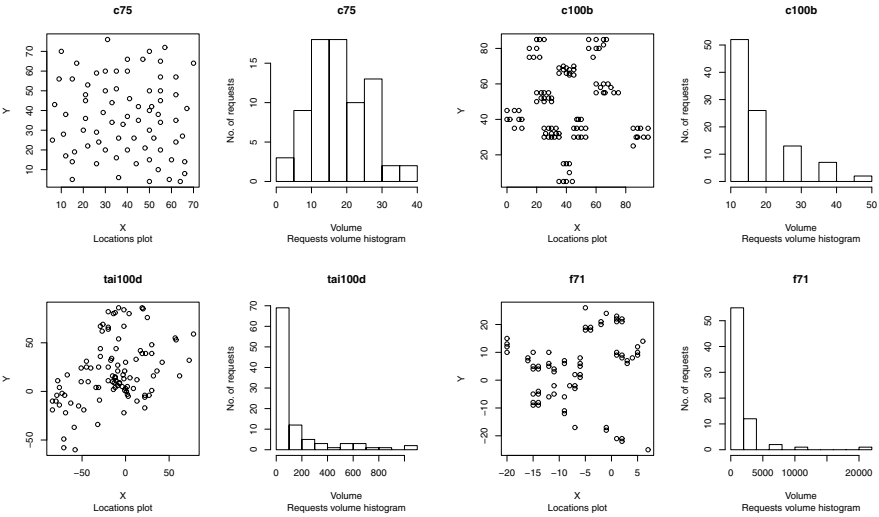


Fig. 3. Benchmark problems examples with clients' requests having spatially uniform location and symmetric volume distributions (c75), spatially clustered location and skewed volume distributions (c100b), spatially semiclustered location and largely skewed volume distributions (tai100d), and spatially clustered location and largely skewed volume distributions (f71)

number of runs on benchmark problem's instances. The other parameters were left at their suggested values (from the literature).

6 Discussion and Conclusions

The DVRP discussed in this paper is an important problem not only because of its real world applications but also a possibility to study continuous optimization algorithms in solving dynamic combinatorial tasks.

The proposed algorithm achieved better results than other PSO approaches in 6 out of 21 test problems. On Christofides benchmark set, 2-phase PSO best solution was averaged out at 0.96 times the length of the MAPSO and DAPSO best results. On Fisher and Taillard benchmark sets it was 0.99 and 1.06, respectively. The overall result was on average 1.02 times longer in comparison with the best results from 2-phase PSO and other approaches and 1.09 times longer in terms of the average results.

Closer analysis of the clients' requests locations and volume distributions reveals that the proposed algorithm performs better for more spatially clustered benchmarks (e.g. c100b) and for more symmetric volume distributions (e.g. c75).

The algorithm's worst performance on the f71 benchmark set, may be related to the configuration of the largest and the second-largest request in this instance of a DVRP. Those two requests are in a close distance to each other and the joint volume of those requests slightly exceeds the capacity of a vehicle, making

the BPP part of the task hard to solve efficiently. Moreover the largest request is a strong outlier in terms of the request volume. It is more then twice the size of the second-largest request in this particular set. In all other test sets this ratio never exceeds 1.37. The examples of the clients' requests location and volume distributions are presented in Fig. 3.

Overall, the results suggest that the DVRP may be efficiently solved by the PSO even without its hybridization with 2-Opt algorithm.

Further research of the proposed algorithm will include the use of alternative encodings of the clients' assignments, further analysis of the impact of locations and requests distribution on the algorithm's performance, and application of a multi-swarm approach.

Acknowledgements. Study was supported by research fellowship within "Information technologies: Research and their interdisciplinary applications" agreement number POKL.04.01.01-00-051/10-00.

The analysis of the requests distribution within the tests sets has been done with the usage of R[14].

References

1. Standard PSO 2011 (2012), http://www.particleswarm.info/
2. Vrp web (2012), http://neo.lcc.uma.es/radi-aeb/WebVRP/
3. Bellaachia, A., Bari, A.: A flocking based data mining algorithm for detecting outliers in cancer gene expression microarray data. In: 2012 International Conference on Information Retrieval Knowledge Management (CAMP), pp. 305–311 (March 2012)
4. Blackwell, T.: Particle swarm optimization in dynamic environments. Evolutionary Computation in Dynamic and Uncertain Environments 51, 29–49 (2007)
5. Cristian, I.: Trelea: The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters 85(6), 317–325 (2003)
6. Cui, X., Potok, T., Palathingal, P.: Document clustering using particle swarm optimization. In: Proceedings IEEE Swarm Intelligence Symposium, SIS 2005, pp. 185–191 (June 2005)
7. Hanshar, F.T., Ombuki-Berman, B.M.: Dynamic vehicle routing using genetic algorithms. Applied Intelligence 27(1), 89–99 (2007), http://dx.doi.org/10.1007/s10489-006-0033-z
8. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. IV, pp. 1942–1948 (1995)
9. Khouadjia, M., Alba, E., Jourdan, L., Talbi, E.G.: Multi-swarm optimization for dynamic combinatorial problems: A case study on dynamic vehicle routing problem. In: Dorigo, M., Birattari, M., Di Caro, G.A., Doursat, R., Engelbrecht, A.P., Floreano, D., Gambardella, L.M., Groß, R., Şahin, E., Sayama, H., Stützle, T. (eds.) ANTS 2010. LNCS, vol. 6234, pp. 227–238. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-15461-4_20
10. Khouadjia, M.R., Sarasola, B., Alba, E., Jourdan, L., Talbi, E.G.: A comparative study between dynamic adapted pso and vns for the vehicle routing problem with dynamic requests. Applied Soft Computing 12(4), 1426–1439 (2012), http://www.sciencedirect.com/science/article/pii/S1568494611004339

11. Kilby, P., Prosser, P., Shaw, P.: Dynamic vrps: A study of scenarios (1998), http://www.cs.strath.ac.uk/~apes/apereports.html
12. Monson, C.K., Seppi, K.D.: Exposing origin-seeking bias in pso. In: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, GECCO 2005, pp. 241–248. ACM, New York (2005), http://doi.acm.org/10.1145/1068009.1068045
13. Pankratz, G., Krypczyk, V.: Benchmark data sets for dynamic vehicle routing problems (2009), http://www.fernuni-hagen.de/WINF/inhalte/benchmark_data.htm
14. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria ISBN 3-900051-07-0 (2012), http://www.R-project.org/
15. Montemanni, R., Gambardella, L., Rizzoli, A., Donati, A.: A new algorithm for a dynamic vehicle routing problem based on ant colony system. Journal of Combinatorial Optimization 10, 327–343 (2005)
16. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. SIGGRAPH Comput. Graph. 21(4), 25–34 (1987), http://doi.acm.org/10.1145/37402.37406
17. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp. 69–73 (1998)
18. Shi, Y., Eberhart, R.: Parameter selection in particle swarm optimization. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, p. 591. Springer, Heidelberg (1998)
19. Spears, W.M., Green, D.T., Spears, D.F.: Biases in particle swarm optimization. International Journal of Swarm Intelligence Research 1(2), 34–57 (2010)

Distribution Network Characteristics in Rough Set Approach

Henryk Piech and Aleksandra Ptak

Czestochowa University of Technology,
Dabrowskiego 73
h.piech@adm.pcz.czest.pl

Abstract. Distribution network optimizing includes a set of issues, such as shortest path DA, PDM, minimal spanning tree MST, the maximum flow MF, cheapest BG flow most numerous association of CMP, the traveling salesman problem TSP, etc.[1, 3, 5, 7–9]. In most of practical implementations of these problems distribution network play a crucial role that is, individual connections (segments between nodes) and their characteristics. These connections should be kept in proper readiness, especially those that are most exploited, that means probably those, the use of which brings the greatest benefits or for other reasons, their exploitation is reasonable. The aim of study is the categorization of the validity of connections in distribution networks.

Keywords: distribution networks, rough set theory, optimization.

1 Introduction

One of the most effective methods of categorization is the use of rough sets convention [15, 16].This approach in the classic version is based on building area of the lower and upper approximation and the negative and boundary area. In our case we dispose complex structures of objects that are parts of the distribution network creating routes which are a sequence of connected (but not for every problem) episodes. The most important parameter influencing the decision on the selection of the category [3, 14] for the specific route is its effectiveness. Evaluation of effectiveness is not a complex issue. We may at once define from what it will depend. Let us mention the number of deliveries along stretches of the route, the value of supplied goods, the frequency of deliveries, etc.- these factors are complementary and even interdependent from each other [2]. Mentioning them we only consider various forms of data structures used for analysis. In the proposed approach one can consider a single, multiple and mixed complexity of the problematic aspect which is the representation of the real situations. The single complexity is limited by a single use of the algorithm. For example, using a disposable minimum spanning tree algorithm to schedule different tasks of supplies transport. Routes will be covered in whole or in part. This will provide a basis for evaluating the effectiveness of their use. Another time, using repeatedly the shortest path algorithm for different nodes (shipping and receiving) one deals

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 559–568, 2013.

with the multiple complexity of the analysis. And finally, when the situation will be based on the use of different types of algorithms, we are dealing with the mixed complexity of the problem. The problematic aspect of the complexity classified that way applies only to data preparation stage. The adequate stage of categorization, however, is treated as the use of these data and only there one applies the convention of rough sets.

2 Additional Characteristics of Distribution Networks Optimization

Optimization of distribution networks primarily concerns flow, therefore supplies to each node from sources or places of manufacture or storage. Algorithms of minimum spanning tree MST, Dijkstra, Simplex, Ford-Fulkerson etc. are commonly used. The main parameter is the bandwidth treated either as a deterministic or uncertain data (interval, fuzzy or approximate) [15, 20]. Results obtained from network optimization methods are used for estimating the cost of their maintaining. Detailed analysis gives the ability to track additional bandwidth parameters such as intensity of use, variability, bottlenecks, seasonality, desirability of further modifications, structural changes, etc. Extending this analysis often involves the addition of the time aspect, the multiplication of sources and loads, the schedule of supplies intensity, interlocking of supplies with different structures. In real situations is therefore necessary to integrate different methods of distribution networks optimization [13, 14]. Multidimensional structure of rough sets allow to create pre-distributions of various parameters and, consequently, seek for objective methods of their integration. One can also apply various types of network cross sections including the similarity of elements attributes, in other words nodes or connections in the network. Additional parameters are considered in cases where they affect the categorization of elements of network structure. Entered parameters often indicate the effectiveness of distribution. They are created on the basis of the financial benefits resulting from the realization of distribution strategy as well as fixed and variable costs. We propose the inclusion in the analysis of orders structures, and perhaps also the frequency of implementation of specific orders. The term structure of the order means configuration of suppliers, receivers and also use of distribution network connection. For example, optimizing the structure of connections with the minimum spanning tree MST we obtain the system of connections that does not include real orders, the delivery sequence of goods for several loads during single supply, the possibility of returning to the source by the shortest route, etc. Similarly, implementing the Ford - Fulkerson algorithm we do not include deliveries to different locations, so it requires additional testing and aggregation of obtained data in terms of exploitation effectiveness of selected network structures [18, 19]. Benefits of using part of the distribution network are related to the supply implementation after connections of this part of the structure. The value of delivery, the frequency of its implementation are factors that directly increase effectiveness of the connection use. Fixed costs of the connection maintenance

and transportation costs after the given section, are naturally factors that reduce the effectiveness of connection exploitation.

$$ef(ei) = 1/n_i/maxV_i \sum_{j=1}^{n_i} V_{i,j}(1 - cv_{i,j}) - ks_i, \tag{1}$$

where
$V_{i,j}$ - value of the supply,
n_i - number of supply with connection e_i,
$maxV_i$ - maximal supply realized with the section e_i,
$cv_{i,j}$ - transportation unit cost (per unit of currency of the good delivery),
k_s - fixed costs to maintain the connection.
The effectiveness of supply routes will therefore be the composition of the effectiveness of included connections:

$$efr(t) = 1/lc \sum_{i=1}^{lc} part(i,j) * ef(e_i), \tag{2}$$

where
lc - number of connections in the network,
$part(i,t) \in \{0,1\}$ - verification of inclusion the connection e_i to the route t.

Introducing forecasts for the size and intensity of supplies one can determine the lower and upper limit of effectiveness exploitation of the route: $\underline{efr(t)}, \overline{efr(t)}$. For each route and its effectiveness, ranges of exploitation will overlap. (Fig.1) Effectiveness ranges that overlap create a categorization dilemma. Designated categories are built on the currently obtained information on supplies, and their expanded ranges based on prediction of supplies.

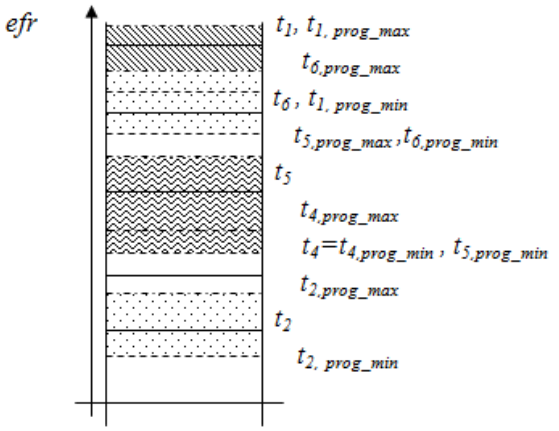


Fig. 1. Distribution of the effectiveness of supply routes, tp_{prog_min}, $tp_{prog_{max}}$ - limits of the effectiveness of supplies forecasts

3 Categorization of Routes Based on Assumptions of the Rough Sets Theory

The problem of assigning to the specific category occurs when scopes overlap. Then one can build a base of reducts [16], that is the set with minimal number of parameters to separate categories. For example, when two or more supply routes create a picture of effective supplies predictions presented in figure 2 than our task can be the selection or planning the organization strategy of cooperation with selected customers. Approximation area of categorization is determined by the interval $[min_{v \in Tc}\{efr(v)\}, max_{u \in Tc}\{efr(u)\}]$, where is found the set of compared supply routes. Let us divide this area into the following sub-ranges:

$$efr(t_i) \subseteq [t_{j,prog\_min}, t_i],$$
$$efr(t_j) \subseteq [t_i, t_{k,prog\_max}],$$
$$efr(t_k) \subseteq [t_i, t_{k,prog\_max}].$$

Designated categories are routes codes: i, j, k. Now we create a zone of shared predictions, that means categories approximations. If $efr(t) \subseteq [t_{j,prog\_min}, t_{k,prog\_max}]$ is the zone of common predictions coincides with the area in which the tested route is localized $t : [t_{j,prog\_min}, t_{k,prog\_max}]$. Now we make a decision on the selection of one of categories. Membership in the category determines the route of delivery $t_p = \{e_p(1), e_p(2), ..., e_p(lt_p)\}$, lt_p - number of connections (sections) of the route t_p. The minimal characteristics of the route, clearly defining the category is called the reduct $r_p = \{e_p(k), e_p(k+1), ..., e_p(k+v) \Rightarrow p : v \rightarrow \min\}$. This projection must have an isomorphic character: $r_p \Rightarrow p$ and $p \Rightarrow r_p$. It is also the shortest connection structure. On the basis of the reduct one can define the delivery category $c = \{p : efr(t)[t_{p,prog\_min}, t_{p,prog\_max}], r_p\}$. Reduct, in our problem, has the sequential character, because the order of route sections may play a decision role (Fig. 3). Sections of reducts are not necessarily linked. Searching for reducts are operations on sequences. They can be started by finding the longest common sequence [6, 17], among those forecasts, which with its scope include the given value of the delivery effectiveness. Subsequently among remaining routes sections we choose two that form the course of delivery. But if there is only one section of the route left, we select other from the common sequence. This heuristic approach, however, does not guarantee uniqueness of the decision on the category selection. It works in situations where areas of forecasts overlap for a small number of routes with common sections. A simpler way to find reducts of the given route t_i is to look for sections of the test route that do not overlap with sections of other routes of the common forecasts zone: $Se(t_i) = \{e_i(k), e_i(k+1), ..., e_i(k+u) \notin t_v \neq t_i,\ \forall v(efr(t_r) \subseteq [t_{p,prog\_min}, t_{p,prog\_max}]),\ efr(t_i) \subseteq [t_{p,prog\_min}, t_{p,prog\_max}],\ u \rightarrow \max\}$.

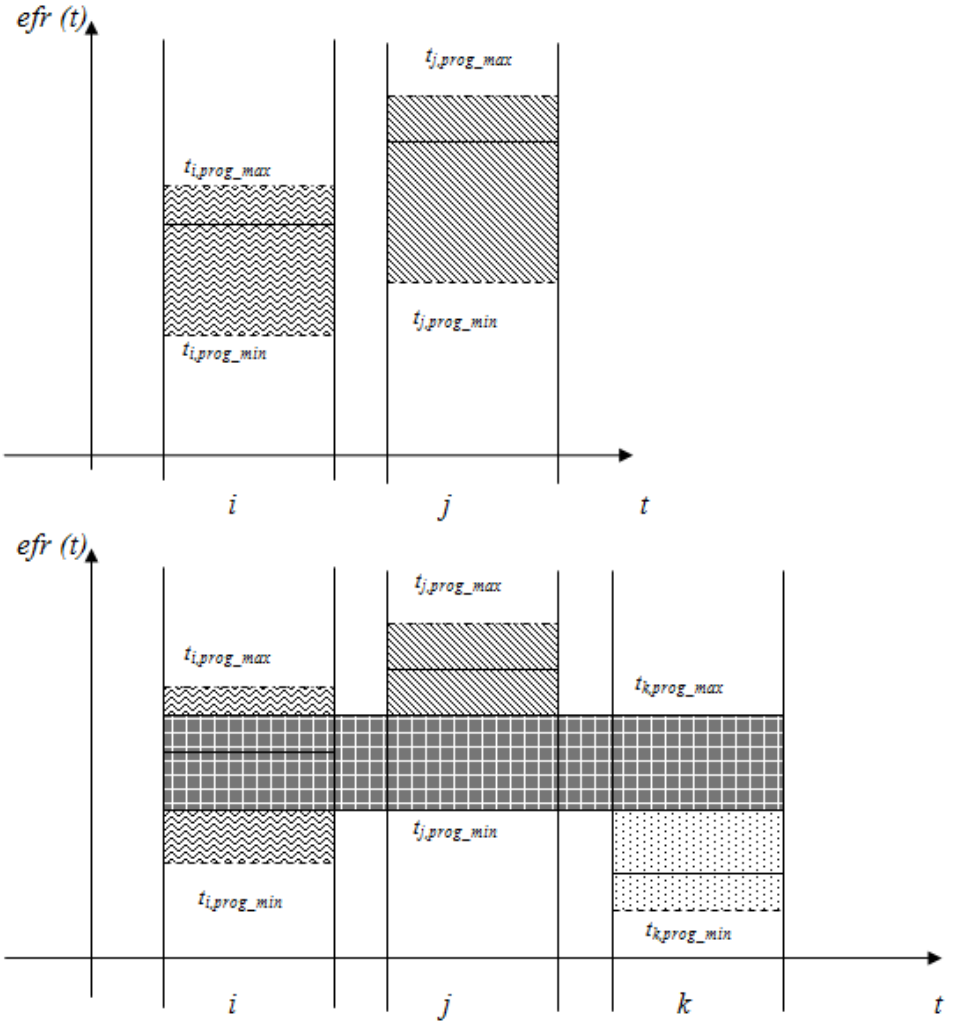


Fig. 2. Examples of effectiveness overlap of the predicted supplies for two or three supply routes

Lemma

Lack of outstanding sections of the delivery route t_i together with the other routes belonging to the same forecasts zone $u < 0$, brings the maximal reduct of the given route to all sections of this route: $r_i = \{e_i(1), ..., e_i(lt_i)\}$, arranged in the sequence of the realization of supply transportation, where lt_i is the number of sections that route t_i consists of.

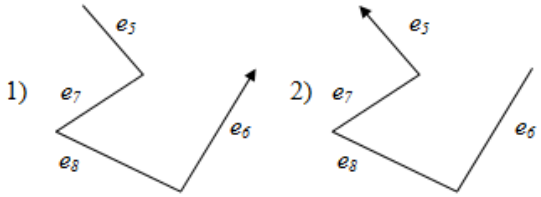


Fig. 3. Supply routes with different reducts $r1 = \{e5, e6\}$ and $r2 = \{e6, e5\}$

Proof

The rejection of the trivial case: the area of common prediction refers only to the test route ti: we assume that there is

$Se*(t_i) = \{e_i(k), e_i(k+1), ..., e_i(k+u) \notin t_v, \forall v(efr(t_r) \subseteq [t_{p,prog\_min}, t_{p,prog\_max}]),$
$efr(t_i) \subseteq [t_{p,prog\_min}, t_{p,prog\_max}], u \to \max\},$

hence

$u = lt_i - 1$ and $t_v = t_i$ that is the route t_v, t_i not differ in any element, which is contrary to assumptions for the creation of reducts, in this case to determine the category value $efr(t)$ is sufficient. Assuming that the condition $t_v \notin t_i$, $Se * (t_i) = \emptyset$ must be fulfilled.

Intermediate case, when the area of the common prediction refers to $lt_i + 1$ of different routes and

$Se * *(t_i) = \{e_i(k), e_i(k+1), ..., e_i(k+u) \notin t_v \neq t_i,$
$\forall v(efr(t_r) \subseteq [t_{p,prog\_min}, t_{p,prog\_max}]), efr(t_i) \subseteq [t_{p,prog\_min}, t_{p,prog\_max}],$
$u \to \max\},$

and also

$u = lt_i - 1$ and then

$Se * *(t_i) = \{e_i(k) \notin t_1 \neq t_i,$
$\forall v(efr(t_r) \subseteq [t_{p,prog\_min}, t_{p,prog\_max}]), efr(t_i) \subseteq [t_{p,prog\_min}, t_{p,prog\_max}],$
$u = 1\}.$
$Se * *(t_i) = \{e_i(k) \notin t_2 \neq t_i,$
$\forall v(efr(t_r) \subseteq [t_{p,prog\_min}, t_{p,prog\_max}]), efr(t_i) \subseteq [t_{p,prog\_min}, t_{p,prog\_max}],$
$u = 1\},$
.....
$Se * *(t_i) = \{e_i(k) \notin t_{lt_i} \neq t_i,$
$\forall v(efr(t_r) \subseteq [t_{p,prog\_min}, t_{p,prog\_max}]), efr(t_i) \subseteq [t_{p,prog\_min}, t_{p,prog\_max}],$
$u = 1\},$

that means any other route which contains one section common with the test route number $lt_i + 1$.

In this situation, we need just two sections to identify each route: the first and last. The extreme case which is the permutation of common connections for $2^{lt_i} - 1$ of different routes. In this case, we can have from 0 to lt_i common sections with the test route. They are, therefore, following situations:

- 0 common connections distinctive $efr(t)$,
- 1 common connections distinctive 2 connections,
- 2 common connections distinctive 3 connections,
.....
- lt_i- common connections distinctive lt_i connections.

The maximal reduct which is the distinctive structure are all sections of the route t_i. When we analyze the new route its categorization may be done by using the maximal number (or length) of sections common with base sections. Taking into account bi-direction of each section one can provide the structure of tested routes with the aid of the diagram similar to the one in Figure 4. On

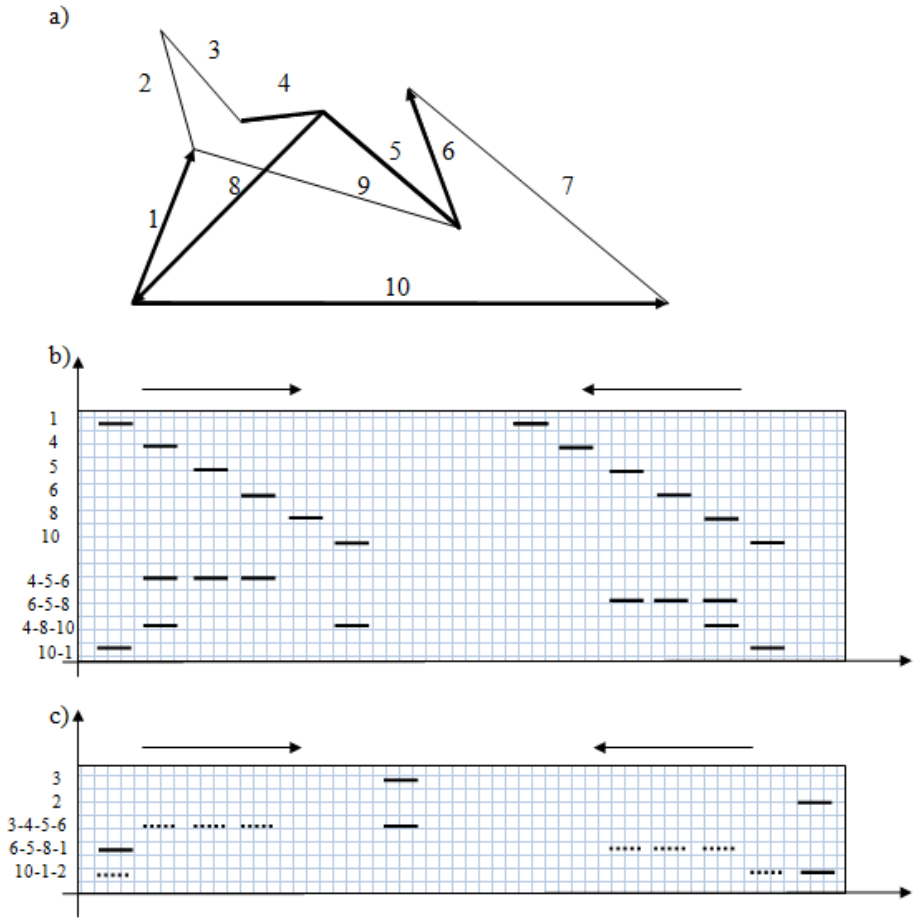


Fig. 4. Elements of new routes categorization (a) the base structure of the category trails (b) the diagram of base routes segments (upper part) and the picture of their use for routes categories (lower part) (c) presentation of new routes with preference (dotted line) segments common with base routes

top of fig.4a the distribution network and the base supply routes are graphically illustrated. The term "base" comes from the fact that they determine categories of validity evaluations of routes depending on the benefits we get from their exploitation. This affects the cost of maintaining them in the state of particular readiness. These are therefore four routes described in the central part of fig.4b and consist of segments depicted above. These sections during the course of delivery are characterized by direction. Hence, the left and right parts of the diagram show directions of transportation during the delivery. The lower part of fig.4c is the presentation of new routes and obtaining characteristics for their categorization that is defining the scale of compliance with the existing base of supply routes. This compatibility is illustrated as the number of common segments NCC, which, for example, for the base route $\{4, 5, 6\}$ and new $3, 4, 5, 6$ equals $\mathrm{NCC}(t_4, t_{n_1}) = 3$. Routes of categories (base) shown in fig. 4 are the basis for designation of these categories (fig. 5). They differ in the effectiveness of their use. Delivering goods along these routes one should remember about the possibility of changing the size and frequency of deliveries. We consider this fact when creating forecasts. Without changing the configuration of the route with the change in the level of supplies the effectiveness of exploitation of the route $(efr(t) = [t_{prog\_max}, t_{prog\_min}])$ will change. The effectiveness of the new route $\{3, 4, 5, 6\}$ lies between $efr(t_{4,prog\_min})$ and $efr(t_{(3,prog\_max)})$. For the categorization scales of compliance $\mathrm{NCC}(t_4, t_{n_1}) = 3$ and $\mathrm{NCC}(t_3, t_{n_1}) = 2$ are tested. The final decision on the categorization is therefore as follows: the new route $\{3, 4, 5, 6\}$ is assigned to the fourth category. With a large-scale divergence of similarities of the new and base path, the problem arises to create a new category. However, this requires further analysis because it is usually associated with additional costs.

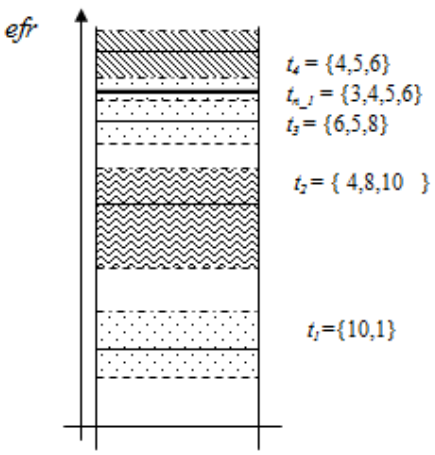


Fig. 5. The effectiveness of distribution network base routes from fig.4a, with prognostic intervals

4 Conclusion

The proposed method of routes categorization brings us closer to real situations and to multi -attribute, according to practical terms, exploitation of distribution networks or their selected parts. In addition, there were taken into account possibilities of changes in volume of supplies, possibilities of new routes introduction and even new categories for their evaluation have been created. Creating new categories may not always be acceptable. Taking into account different modes of exploitation can be forward looking. Nevertheless, such generalization seems to be intentional, because it suggests the possibility of using the method in a wider range of problems and a high level of flexibility in decision making. The forecast of the size of future supplies also should be more balanced and realistic. This requires further analysis and additional data. Generally speaking, all additional procedures are possible to realize, as in other tasks they still are implemented.

References

1. Burkard, R.E., Derigs, U.: Assignment and Matching Problems. Lecture Notes in Econom. and Math. Systems, vol. 184. Springer, New York (1980)
2. Cheriton, D., Tarjan, R.E.: Finding minimum spanning trees. SIAM J. Comput. 5, 724–742 (1976)
3. Cheung, T.Y.: Computational comparison of eights methods for the maximum network flow problem. ACM Trans Math. Software 6, 1–16 (1980)
4. Crowder, H., Padberg, M.W.: Solving large scale symmetric traveling salesman problems to optimality. Management Sci. 26, 495–509 (1980)
5. Denardo, E.V., Fox, L.B.: Shortest - route methods: Reaching, pruning and buckets. Oper. Res. 27, 215–248 (1979)
6. Deo, N., Pang, C.Y.: Shortest path algorithms: taxonomy and annotation. Networks 14, 275–323 (1984)
7. Derigs, U., Heske, A.: A computational study on some methods for solving the cardinality matching problem. Angew. Inform. Jg. 22, 249–254 (1980)
8. Ford, L.R.: Network flow theory. Rand Corporation, Santa Monica (1956)
9. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Canad. J. Math. 8, 399–404 (1956)
10. Galil, Z.: On the theoretical efficiency o f various network flow algorithms. Theoret. Comput. Sci. 14, 103–111 (1981)
11. Glover, F., Klingman, D., Mote, J., Whitman, D.: Comprehensive computer evaluation and enhancement of maximum flow algorithms, Research. Report, 356, Center for Cybernetic Studies, University of Texas, Austin (1979)
12. Golden, B., Bodin, L., Doyle, T., Steward, W.: Approximate travelling salesman algorithms. Oper. Res. 28, 694–711 (1980)
13. Kennington, J.L., Helgason, R.V.: Algorithms for network programming. Wiley-Interscience, New York (1980)
14. Minieka, E.: Optimization Algorithms for Networks and Matroids. Holt, Rinehart and Winston, New York (1978)
15. Pawlak, Z.: Rough Sets and Fuzzy Sets. Fuzzy Sets and Systems 17, 99–102 (1985)
16. Pawlak, Z.: Rough Sets. International Journal of Information and Computer Science 11, 341–356 (1982)

17. Pierce, A.R.: Bibliography on algorithms for shortest path, shortest spanning tree and related circuit routing problems. Networks 5, 129–149 (1975)
18. Tadeusiewicz, R.: Introduction to Inteligent Systems. In: Wilamowski, B.M., Irvin, J.D. (eds.) The Industrial Electronic Handbook, ch. 1, pp. 1-1 – 1-12. CRC Press, Boca Raton (2011)
19. Tadeusiewicz, R.: Place and role of Intelligence Systems in Computer Science. Computer Methodsin Material Science 10(4), 193–206 (2010)
20. Zadeh, N.: Near - equivalence of network flow algorithms, Technical Report, 26, Dept. Operations Research, Stanford University, CA (1979)

Comparison between PSO and AIS on the Basis of Identification of Material Constants in Piezoelectrics

Arkadiusz Poteralski[1], Mirosław Szczepanik[1], Grzegorz Dziatkiewicz[1], Wacław Kuś[1], and Tadeusz Burczyński[1,2]

[1] Silesian University of Technology, Gliwice, Poland
{arkadiusz.poteralski,miroslaw.szczepanik,
grzegorz.dziatkiewicz,waclaw.kus}@polsl.pl
[2] Cracow University of Technology, Cracow, Poland
tburczyn@pk.edu.pl

Abstract. The paper deals with an application of the artificial immune system (AIS) and particle swarm optimizer (PSO) to the identification problem of piezoelectric structures analyzed by the boundary element method (BEM). The AIS and PSO is applied to identify material properties of piezoelectrics. The AIS is a computational adaptive system inspired by the principles, processes and mechanisms of biological immune systems. The algorithms typically use the characteristics of the immune systems like learning and memory to simulate and solve a problem in a computational manner. The PSO algorithm is based on the models of the animals social behaviours: moving and living in the groups. PSO algorithm realizes directed motion of the particles in n-dimensional space to search for solution for n-variable optimisation problem.The main advantage of the bioinspired methods (AIS and PSO), contrary to gradient methods of optimization, is the fact that it does not need any information about the gradient of fitness function.

Keywords: artificial immune system, particle swarm optimizer, boundary element method, identification, piezoelectricity.

1 Introduction

Identification of parameters in physical systems using the artificial intelligence techniques is a very active field of research [3, 12, 14, 15, 19, 22, 27, 31]. The main approach in solving the identification problem is to find the parameters of physical systems using the response of the structure. Several novel techniques have been recently developed. One of the best known approaches is based on the minimization of the special functional [13]. In [13], the identification problem consists in finding such values of the material constants, which give solution fields that differ the least from the measured ones known from the experiment. The fields on the boundary are known at a given number of boundary points. The identification problem of material constants is equivalent to the minimization problem of the special functional. Minimization with respect to the searched parameters can be done using many approaches.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 569–581, 2013.

A general approach is presented in [29]. In this work the genetic least-squares technique is applied for system identification. In [31] genetic algorithms are used for identification of material constants in anisotropic media. The evolutionary algorithm for identification of material coefficients of human pelvic bone was used in [6]. In [16], the authors applied the evolutionary-simplex algorithm and structural dynamic response to identify the structural physical parameters. A genetic algorithm was also applied in [27] to search for the elastic parametres of composites. In another interesting approach the genetic or evolutionary algorithms are connected with other optimization techniques [14]. In [14], the combination of topological sensitivity and genetic algorithms are used for identification inverse problems in anisotropic materials. Also [13] proposed gradient-based, genetic and hybrid optimization algorithms for material parameters identification problem. Identification of effective elastic constants of composite plates based on a hybrid genetic algorithm was also considered in [19].

Another approach to inverse identification problems, based on hybrid techniques, has been developed in [5], where the surrogate-model accelerated random search algorithm for global optimization was applied to inverse material identification. Recently, [32] proposed the immune evolutionary algorithm incorporating chaos optimization. An evolutionary AIS was applied to multi-objective optimization in [28].

In this article new method of identification problem is proposed. This article concerns the application of the boundary element method (BEM) [4] coupled with a artificial immune system (AIS) [30] and particle swarm optimizer (PSO) [21] in order to identify the material properties of piezoelectrics. This method is used in other interesting approach [23]. The authors also developed another global optimization method like evolutionary algorithm (EA) [7, 8, 9, 10, 11].

In piezoelectric materials, mechanical and electric fields are coupled, namely, they produce an electric field when deformed or conversely, they deform when subjected to an electric field. Piezoelectric materials are widely used as sensors and actuators in smart structures and micro-electro-mechanical systems (MEMS). In [1], authors developed a finite element model for the identification of mechanical and piezoelectric properties through gradient optimization and experimental vibration data. In [2], authors estimated piezoelastic and viscoelastic properties in lamined structures by using the classical algorithms of nonlinear optimization combined with the sensitivities and the finite element model. The piezoelectricity phenomenon is widely used in many devices, for example, sensors, actuators, MEMS and transducers. The ceramic piezoelectrics are solids, which belong to the hexagonal symmetry class of crystals. These crystalic solids have anisotropic physical properties, therefore in the paper a homogeneous, transversal isotropic, linear elastic and dielectric model of the piezoelectric material is chosen. For this material, the piezoelectric effect in the two-dimensional case is described by nine material constants: four elastic, three piezoelectric and two dielectric. In addition, the dielectric constants are twenty orders of magnitude smaller, than the elastic constants. These facts, i.e. a relatively big number of the constants and lack of gradient information, make the identification problem of the piezoelectric material constants quite complicated. An analysis of piezoelectric devices requires a solution of the coupled mechanical and electrical partial differential equations. In this paper the BEM is implemented to solve the static two-dimensional problem of linear piezoelectricity. To obtain the fundamental solutions, the Stroh formalism is used. To solve both direct and inverse problems of piezoelectricity, the BEM computer

code is developed and the AIS and PSO algorithm is coupled with the BEM code. Numerical examples are given and good results are obtained.

In the authors' opinion, no application of the AIS or PSO for solving inverse identification problems in piezoelectrics modelled by using the BEM can be found in the literature until now.

2 Artificial Immune Systems

The artificial immune systems are developed on the basis of a mechanism discovered in biological immune systems [25]. An immune system is a complex system which contains distributed groups of specialized cells and organs. The main purpose of the immune system is to recognize and destroy pathogens - funguses, viruses, bacteria and improper functioning cells.

The artificial immune systems take only a few elements from the biological immune systems. The most frequently used are the mutation of the B-cells, proliferation, memory cells, and recognition by using the B and T-cells. The presented approach is based on the Wierzchoń's algorithm [30], but the mutation operator is changed. The Gaussian mutation is used instead of the nonuniform mutation in the presented approach. At the beginning of the AIS the memory cells are created randomly. They proliferate and mutate creating B-cells. The number of clones created by each memory cell is determined by the memory cells objective function value. The objective functions for B-cells are evaluated. The selection process exchanges some memory cells for better B-cells. The selection is performed on the basis of the geometrical distance between each memory cell and B-cells (measured by using design variables). The crowding mechanism removes similar memory cells. The similarity is also determined as the geometrical distance between memory cells. The process is iteratively repeated until a stop condition is fulfilled. An AIS is schematically shown in Fig. 1a).

3 The Particle Swarm Optimizer

The particle swarm algorithms [21], similarly to the evolutionary and immune algorithms, are developed on the basis of the mechanisms discovered in the nature. The swarm algorithms are based on the models of the animals social behaviours: moving and living in the groups. The results of this biological examination were used by Kennedy and Eberhart [20], who proposed Particle Swarm Optimiser – PSO [18, 26]. This algorithm realizes directed motion of the particles in n-dimensional space to search for solution for n-variable optimisation problem. PSO works in an iterative way. The location of one individual (particle) is determined on the basis of its earlier experience and experience of whole group (swarm). Moreover, the ability to memorize and, in consequence, returning to the areas with convenient properties, known earlier, enables adaptation of the particles to the life environment. The optimisation process using PSO is based on finding the better and better locations in the search-space (in the natural environment that are for example hatching or feeding grounds). The algorithm with continuous representation of design variables and constant constriction coefficient (constricted continuous PSO) has been used in presented research. In this

approach each particle oscillates in the search space between its previous best position and the best position of its neighbours, with expectation to find new best locations on its trajectory. When the swarm is rather small (swarm consists of several or tens particles) it can be assumed that all the particles stay in neighbourhood with currently considered one. In this case we can assume the global neighbourhood version and the best location found by swarm so far is taken into account – current position of the swarm leader. The position of the i-th particle is changed by stochastic velocity v_i, which is dependent on the particle distance from its earlier best position and position of the swarm leader. This approach is given by the following equations:

$$v_{ij}(k+1) = wv_{ij}(k) + \phi_{1j}(k)\left[q_{ij}(k) - d_{ij}(k)\right] + \phi_{2j}(k)\left[\hat{q}_{ij}(k) - d_{ij}(k)\right] \tag{1}$$

$$d_{ij}(k+1) = d_{ij}(k) + v_{ij}(k+1), \quad i = 1,2,...,m \; ; \; j = 1,2,...,n \tag{2}$$

where:
$\phi_{1j}(k) = c_1 r_{1j}(k)$; $\phi_{2j}(k) = c_2 r_{2j}(k)$, m – number of the particles, n – number of design variables (problem dimension), w – inertia weight, c_1, c_2 – acceleration coefficients, r_1, r_2 – random numbers with uniform distribution [0,1], $d_i(k)$ – position of the i-th particle in k-th iteration step, $v_i(k)$ – velocity of the i-th particle in k-th iteration step, $q_i(k)$ – the best found position of the i-th particle found so far, $\hat{q}_i(k)$ – the best position found so far by swarm – the position of the swarm leader, k – iteration step.

The velocity of the i-th particle is determined by three components of the sum in Equation (1). The first component $wv_i(k)$ plays the role of the constraint to avoid excessive oscillation in the search space. The inertia weight w controls the influence of particle velocity from the previous step on the current one. In this way this factor controls the exploration and exploitation. Higher value of inertia weight facilitates the global searching, and lower – the local searching. The inertia weight plays the role of the constraint applied for the velocities to avoid particles dispersion and guaranteeing convergence of the optimization process. The second component $\phi_1(k)\left[q_i(k) - d_i(k)\right]$ realizes the cognitive aspect. This component represents the particle distance from its best position found earlier. It is related to the natural inclination of the individuals (particles) to the environments where they had the best experiences (the best value of the fitness function). The third component $\phi_2(k)\left[\hat{q}_i(k) - d_i(k)\right]$ represents the particle distance from the position of the swarm leader. It refers to the natural inclination of the individuals to follow the other which achieved a success. The flowchart of the particle swarm optimizer is presented in Fig. 1b. At the beginning of the algorithm the particle swarm of assumed size is created randomly. Starting positions and velocities of the particles are created randomly. The objective function values are evaluated for each particle. In the next step the best positions of the particles are updated and the swarm leader is chosen. Then the particles velocities are modified by means of the Equation (1) and particles positions are modified according to the Equation (2). The process is iteratively repeated until the stop condition

is fulfilled. The stop condition is typically expressed as the maximum number of iterations. The general effect is that each particle oscillates in the search space between its previous best position (position with the best fitness function value) and the best position of its best neighbour (relatively swarm leader), hopefully finding new best positions (solutions) on its trajectory, what in whole swarm sense leads to the optimal solution.

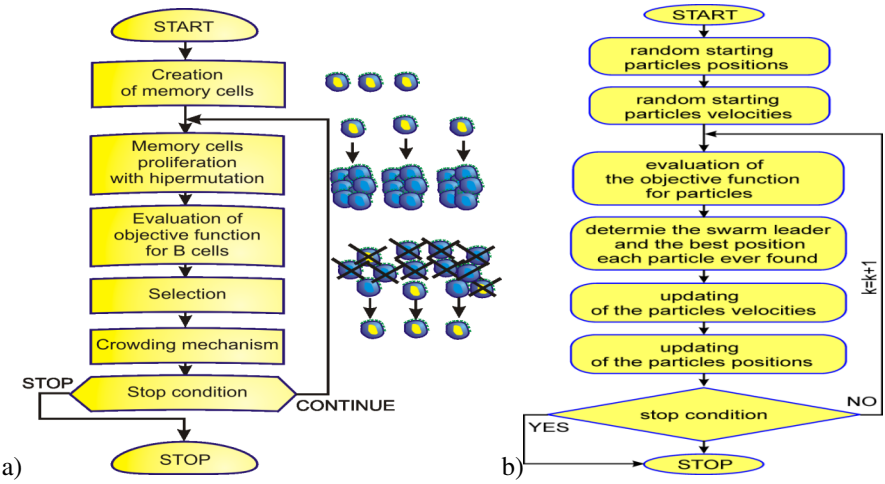


Fig. 1. Bioinspired algorithms: a) artificial immune system, b) particle swarm optimiser

4 Piezoelectricity Problem

An analysis of piezoelectric devices requires a solution of coupled mechanical and electrical partial differential equations [17], [24]. In this paper the boundary element method (BEM) [4] is implemented to solve the static two-dimensional problem of linear piezoelectricity.

The coupled field equations of static piezoelectricity are given by the following system of partial differential equations [17], [24]:

$$C_{ijkl}u_{k,li} + e_{lij}\phi_{,li} = -f_j \qquad e_{ikl}u_{k,li} - \varepsilon_{il}\phi_{,li} = 0 \tag{3}$$

The tensors C_{ijkl}, e_{lij}, ε_{il} denote elastic moduli, measured in a constant electric field, piezoelectric constants and dielectric constants, measured at constant strains, respectively; u_k is the displacement vector, φ denotes the electric potential; f_j is the body force vector per unit volume. In equations (3) the intrinsic electric charge is neglected, because piezoceramic materials are dielectrics, which possess no free electric charges [17].

To get the classical boundary – value problem formulation, equations (3) must be completed with the boundary conditions, both mechanical and electric:

$$\Gamma_t : t_i = \bar{t}_i \;;\; \Gamma_u : u_i = \bar{u}_i \qquad \Gamma_\phi : \phi = \bar{\phi} \;;\; \Gamma_\omega : \omega = \bar{\omega} \tag{4}$$

In equations (4) t_i denotes the tractions, ω is the charge flux density; Γ_t, Γ_u, Γ_φ and Γ_ω denote parts of the boundary Γ where tractions, displacements, potentials and charge flux densities are prescribed.

The coupled field equations with boundary conditions formulate the direct problem in linear piezoelectricity.

The boundary – value problem of linear piezoelelctricity can be formulated in a much more convenient form using generalized quantities [17], [24]. The following vectors are introduced:

$$U_K = \begin{Bmatrix} u_k \\ \varphi \end{Bmatrix}; \quad T_J = \begin{Bmatrix} t_j \\ \omega \end{Bmatrix}; \quad B_J = \begin{Bmatrix} b_j \\ 0 \end{Bmatrix} \tag{5}$$

where U_K, T_J and B_J are the generalized displacement, traction and body force vector, respectively. Then, the coupled field equations are given by the operator equation:

$$L_{JK} U_K = -B_J \tag{6}$$

where L_{JK} is the 2D elliptic operator of static piezoelectricity.

In the present work a homogeneous, transversal isotropic, linear elastic and dielectric model of the piezoelectric material is chosen. For this model, the operator L_{JK} , in the two - dimensional case, has a form:

$$L_{JK} = \begin{bmatrix} c_{11}\partial_{11} + c_{33}\partial_{22} & (c_{12} + c_{13})\partial_{12} & (e_{21} + e_{13})\partial_{12} \\ & c_{33}\partial_{11} + c_{22}\partial_{22} & e_{13}\partial_{11} + e_{22}\partial_{22} \\ sym & & -\varepsilon_{11}\partial_{11} - \varepsilon_{22}\partial_{22} \end{bmatrix} \tag{7}$$

where ∂_{ij} is a differential operator (differentation with respect to the spatial coordinates). The coefficients c_{ij}, e_{ij} and ε_{ij} are the values of the elastic, piezoelectric and dielectric constants, respectively; and i, j=1,2.

The next step in the BEM formulation is obtaining a reciprocity relation. The system of equations (6) is weighted with a test function and integrated by parts. Then the piezoelectric reciprocity relation is obtained.

When the test function is chosen as a fundamental solution of the static piezoelectric differential operator L_{JK} and a limiting process to the boundary is done, the boundary integral formula is obtained.

In the present work homogeneous, transversal isotropic, linear elastic and dielectric model of the piezoelectric material is chosen [17], [24]. For this material, the piezoelectric effect in the two – dimensional case is described by the nine material constants. These constants are: four elastic constants, three piezoelectric constants and two dielectric constants. In addition, the dielectric constants are twenty orders of the magnitude smaller, than the elastic constants. This fact, i.e. a relatively big number of the constants and lack of gradient information, makes the identification problem of the piezoelectric material constants quite complicated [17].

Since ceramic piezoelectric materials are anisotropic, the fundamental solutions are rather complicated, even for the transversal isotropic model of the material. To obtain the fundamental solutions, the Stroh formalism is used [24]. The Stroh formalism is a powerful and elegant analytic technique for the anisotropic elasticity, which is expanded to the linear piezoelectricity in this case. The orientation of the polarization direction is also taken into account using this formalism. The formalism requires the solution of the special eigenvalue problem with respect to the material constants of the piezoelectric material. The eigenvalues and the eigenvectors related to these constants, are specially transformed according to the polarization direction. To solve approximately the boundary integral equation, the boundary element method is applied. The boundary Γ is divided into boundary elements. The boundary generalized displacements and tractions are approximated using shape functions. In the present method constant boundary elements are used.The discretized equation is applied to all boundary nodes, and the following system of linear algebraic equations is obtained:

$$HU = GT \tag{8}$$

In equation (8) H denotes matrix which depends on derivatives of the fundamental solution, G is a matrix which depends on the fundamental solution. The vectors U and T contain discretized values of the boundary generalized displacements and tractions. The set of equations is modified according to the boundary conditions and solved giving unknown boundary quantities. The modified system of equations has a form:

$$CX = F \tag{9}$$

where C denotes the matrix, which depends on the fundamental solution and its derivatives, F is obtained using boundary conditions and X contains the unknown values of the generalized boundary displacements and tractions.

5 Identification Problem

The identification problem is: how to find the material constants and the polarization direction using the response of the structure [17]? The identification problem consists in finding such values of the material constants, which give the solution fields (both mechanical and electric) that differ the least from the measured ones known from the numerical experiment [17]. The fields on the boundary (such as displacements, tractions, potential and charge flux density) are known in a given number of boundary points called sensors, see in Fig. 2. The identification problem of material constants and polarization direction is equivalent to the minimization problem of the special functional.

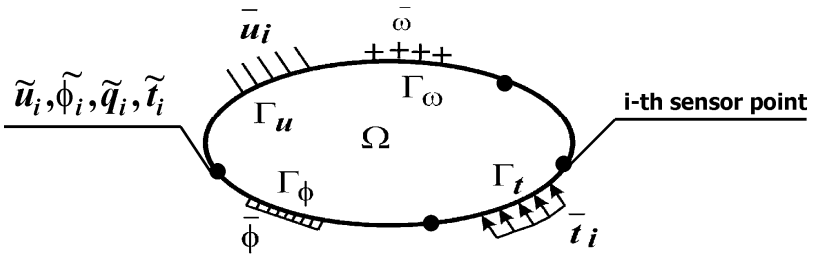


Fig. 2. Sensor points on the boundary

At each sensor point values of displacements $\tilde{u}_i$, potential $\tilde{\phi}_i$, charge flux density $\tilde{q}_i$ and tractions $\tilde{t}_i$ are known. All values of the above quantities are dependent on the material constants and the polarization direction. The geometry of the body does not change, and the boundary conditions are constant. The functional of the identification problem, in this case, has a form [17]:

$$J(d) = w_1 \sum_{i=1}^{nds} \left| u_i(d) - \tilde{u}_i \right| + w_2 \sum_{j=1}^{nps} \left| \varphi_j(d) - \tilde{\varphi}_j \right| + w_3 \sum_{k=1}^{ncs} \left| q_i(d) - \tilde{q}_i \right| + w_4 \sum_{l=1}^{nts} \left| t_l(d) - \tilde{t}_l \right| \quad (10)$$

where $u_i(d)$, $\phi_j(d)$, $q_i(d)$ and $t_i(d)$ denotes displacement, potential, charge flux density and tractions, computed at each sensor point, which are dependent on identified parameters d:

$$d = (c_{ij}, e_{ij}, \varepsilon_{ij}, \theta) \quad (11)$$

where θ denotes the polarization direction.

The functional (10) is also called the fitness function. The optimal value of the fitness function is equal to zero. The fitness function values are evaluated using BEM and also measurements are simulated using BEM. To solve the identification problem the artificial immune system [3, 12, 25, 30] and particle swarm optimizer is used [20, 21, 26]. The solution of coupled field equations using BEM is needed for each B-cell in the AIS or each particle in the PSO algorithm. Minimization of the fitness function with respect to the searched parametrs is done using PSO or AIS optimizer.

The identification of piezoelectric material constants and the polarization direction of a plate is considered. The plate is subjected to a stress in the y direction and an applied voltage as shown in Fig. 3 [17]:

The PZT-4 ceramic material is modelled. The material properties are shown in Table 1. The applied voltage is V_0=1000 V. On the horizontal edges of the strip, the charge flux density is equal to zero. The applied stresses are equal to σ=5 MPa. The length of the strip is equal to L=1 mm, the height is 2h=1 mm. To discretize the boundary of the strip, 20 constant boundary elements are applied.

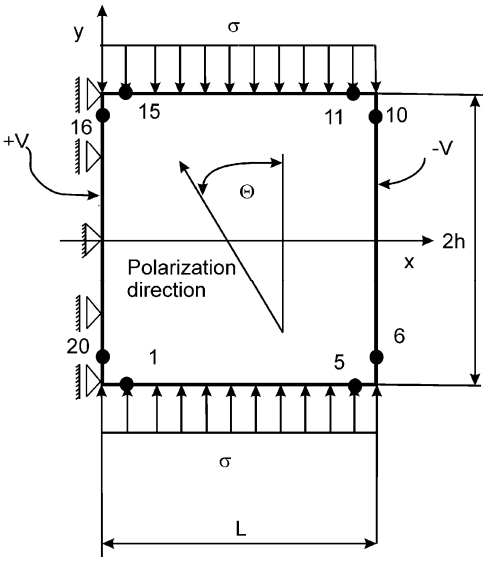


Fig. 3. Geometry and boundary conditions of the plate

Table 1. Values of the material constants

constants								
c_{11} [Pa]	c_{12}	c_{13}	c_{22}	e_{13} [C/m^2]	e_{21}	e_{22}	ε_{11} [C/Vm]	ε_{22}
$0.73 \cdot 10^{11}$	$0.64 \cdot 10^{11}$	$0.21 \cdot 10^{11}$	$0.28 \cdot 10^{11}$	-2.1	19.1	12.3	$8.13 \cdot 10^{-9}$	$143 \cdot 10^{-9}$

Ten design variables are used: 9 material constants and the value of the angle θ, which describes the polarization direction. The parameters of the bioinspired methods are presented in Table 2. The constraints on design parameters values are presented in Table 3.

Table 2. Parameters of bioinspired methods

AIS			
the number of memory cells	the number of the clones	crowding factor	Gaussian mutation
6	30	0.25	0.25
PSO			
Numbers of particles	Inertia weight w	Acceleration coefficient c_1	Acceleration coefficient c_2
40	0.73	1.47	40

The bioinspired optimizers were coupled with the BEM code. The measured values were simulated using numerical solution. The displacements, electric potential and charge sensors are placed in every node. The displacement sensors measure x and y displacements of nodes. The coefficients in the fitness function (10) are as follows: w1=1, w2=10^{-7}, w3=10^{-3}, w4=0.

Table 3. The design parameters constraints

gen	constant	minimum value	maximum value
1	c_{11} [Pa]	$0.2 \cdot 10^{11}$	$0.8 \cdot 10^{11}$
2	c_{12}	$0.2 \cdot 10^{11}$	$0.8 \cdot 10^{11}$
3	c_{13}	$0.2 \cdot 10^{11}$	$0.8 \cdot 10^{11}$
4	c_{22}	$0.2 \cdot 10^{11}$	$0.8 \cdot 10^{11}$
5	e_{13} [C/m^2]	-3.0	25.0
6	e21	-3.0	25.0
7	e22	-3.0	25.0
8	ε_{11} [C/Vm]	5·10-9	150·10-9
9	ε_{22}	5·10-9	150·10-9
10	θ [°]	0	180

In the first identification problem all the material constants without polarization direction are searched. The exact and best found values of material constants after 518th for AIS and 44th for PSO iteration are presented in Table 4:

Table 4. Results of identification of the material constants

constant	exact value	the best after 518 iter. using AIS	error after 518 iter. % using AIS	the best after 44 iter. using PSO	error after 44 iter. % using PSO
c_{11} [Pa]	$0.73 \cdot 10^{11}$	$0.76 \cdot 10^{11}$	4.1	$0.74 \cdot 10^{11}$	1.4
c_{12}	$0.64 \cdot 10^{11}$	$0.61 \cdot 10^{11}$	4.7	$0.69 \cdot 10^{11}$	7.8
c_{13}	$0.21 \cdot 10^{11}$	$0.20 \cdot 10^{11}$	4.8	$0.212 \cdot 10^{11}$	0.9
c_{22}	$0.28 \cdot 10^{11}$	$0.29 \cdot 10^{11}$	3.6	$0.298 \cdot 10^{11}$	6.4
e_{13} [C/m^2]	-2.1	-2.28	8.1	-1.98	5.7
e_{21}	19.1	18.0	1.1	20.1	5.3
e_{22}	12.3	11.2	5.8	11.9	3.3
ε_{11} [C/Vm]	$8.13 \cdot 10^{-9}$	$8.75 \cdot 10^{-9}$	7.6	$7.93 \cdot 10^{-9}$	2.46
ε_{22}	$143 \cdot 10^{-9}$	$143.38 \cdot 10^{-9}$	0.3	$142.76 \cdot 10^{-9}$	0.2

It can be noticed that the material constants are identified with good accuracy, because the present boundary – value problem is quite simple and its solution does not contain many information, which is necessary for the identification. A relatively big number of the constants makes the identification problem quite complicated. In the second identification task the piezoelectric constants are searched. The exact and best found values of piezoelectric material constants after 82nd for AIS and 128th for PSO iteration are presented in Table 5:

Table 5. Results of identification of the piezoelectric constants

constant	exact value	the best after 82 iter. using AIS	error after 82 iter. % using AIS	the best after 128 iter. using PSO	error after 128 iter. % using PSO
e_{13} [C/m^2]	-2.08	-2.06	1.9	-2.08	0.0
e_{21}	19.1	19.10	0.0	19.1	0.0
e_{22}	12.3	12.19	0.9	12.3	0.0

In the third identification problem the piezoelectric constants and polarization direction are searched. The exact and best found values are presented in Table 6. It is easy to see, that the polarization direction and the piezoelectric constants are identified very good. Of course, the accuracy of the solution depends on the boundary – value problem for which, the identification problem is being solved. Also, the parameters of the AIS and PSO algorithm have a great influence on a quality of the results. These parameters are chosen after several tests.

Table 6. Results of identification of the piezoelectric constants and polarization direction

constant	exact value	the best after 444 iter. using AIS	error after 444 iter. % using AIS	the best after 184 iter. using PSO	error after 184 iter. % using PSO
e_{13} [C/m^2]	-2.08	-2.09	0.5	-2.08	0.0
e_{21}	19.1	19.09	0.1	19.1	0.0
e_{22}	12.3	12.38	0.7	12.3	0.0
θ [°]	90	90.00	0.0	90	0.0

6 Conclusions

In this paper, the formulation and application of the BEM and the AIS or PSO to identification of piezoelectrics are presented. The AIS and PSO can be simply implemented because it needs only the values of the objective function.

The AIS or PSO coupled with the BEM give an efficient computational intelligence method for identification of piezoelectric material constants and the polarization direction. To solve both direct and inverse problems of piezoelectricity, the BEM computer code is developed and the AIS and PSO are coupled with the BEM code. Numerical examples are given and good results are obtained. Comparison between PSO and AIS proves good effectiveness of particle swarm optimization method. The results of the numerical examples confirm the efficiency of the proposed optimization method and demonstrate that the method based on particle swarm computation is an effective technique for solving computer aided optimal design problems.

Acknowledgment. The research is partially financed as a research project no. BK-204/RMT4/2012.

References

1. Araujo, A.L., Mota Soares, C.M., Herskovits, J., Pedresen, P.: Development of a finite element model for the identification of mechanical and piezoelectric properties through gradient optimization and experimental vibration data. In: Composite Structures, pp. 307–318 (2002)
2. Araujo, A.L., Mota Soares, C.M., Herskovits, J., Pedresen, P.: Estimation of piezoelastic and visoelastic properties in laminated structures. In: Composite Structures, pp. 168–174 (2009)
3. Balthrop, J., Esponda, F., Forrest, S., Glickman, M.: Coverage and generalization in an artificial immune system. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002, pp. 3–10. Morgan Kaufmann, New York (2002)
4. Brebbia, C.A., Dominguez, J.: Boundary elements. In: An introductory course. Computational Mechanics Publications, McGraw, Hill Book Company, Southampton, Boston (1992)
5. Brigham, J.C., Aquino, W.: Surrogate-model accelerated random search algorithm for global optimization with applications to inverse material identification. Computer Methods in Applied Mechanics and Engineering 196, 4561–4576 (2007)
6. Burczynski, T., John, A., Kuś, W., Orantek, P., Poteralski, A.: The evolutionary algorithm and hipersurface in identification of material coefficients of human pelvic bone. Acta of Bioengineering and Biomechanics 5, 61–66 (2003)
7. Burczyński, T., Poteralski, A., Szczepanik, M.: Genetic generation of 2-D and 3-D structures Second M.I.T. Conference on Computational Fluid and Solid Mechanics Massachusetts Institute of Technology Cambridge, MA 02139 U.S.A
8. Burczyński, T., Poteralski, A., Szczepanik, M.: Topological evolutionary computing in the optimal design of 2D and 3D structures. Engineering Optimization 39(7), 811–830
9. Burczynski, T., Bereta, M., Poteralski, A., Szczepanik, M.: Immune Computing: Intelligent Methodology and Its Applications in Bioengineering and Computational Mechanics. In: Computer Methods In Mechanics, pp. 165–181 (2010)
10. Burczyński, T., Kuś, W., Długosz, A., Poteralski, A., Szczepanik, M.: Sequential and Distributed Evolutionary Computations in Structural Optimization. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 1069–1074. Springer, Heidelberg (2004)
11. Burczynski, T., Dlugosz, A., Kus, W., Orantek, P., Poteralski, A.: Intelligent computing in evolutionary optimal shaping of solids. In: 3rd International Conference on Computing, Communications and Control Technologies, vol. 3, pp. 294–298 (2005)
12. de Castro, L.N., Timmis, J.: Artificial immune iystems as a novel soft computing paradigm. Soft Computing 7(8), 526–544 (2003)
13. Chaparro, B.M., Thullier, S., Menezes, L.F., Manach, P.Y., Fernandes, J.V.: Material parameters identification: Gradient-based, genetic and hybrid optimization. Computational Materials Science 44, 339–346 (2008)
14. Comino, L., Gallego, R., Rus, G.: Combining topological sensitivity and genetic algorithms for identification inverse problems in anisotropic materials. Computational Mechanics 41, 231–242 (2008)
15. Dlugosz, A.: Evolutionary computation in thermoelastic problems. In: IUTAM Symposium on Evolutionary Methods in Mechanics, vol. 117, pp. 69–80 (2004)
16. Du, X.: Zengdi, Structural physical parameter identification based on evolutionary-simplex algorithm and structural dynamic response. Earthquake Engineering and Engineering Vibration 2, 225–236 (2003)

17. Dziatkiewicz, G., Kuś, W., Burczyński, T., Fedeliński, P.: Identification of piezoelectric material constants using distributed evolutionary algorithm. In: Methods of Artificial Intelligence, AI-METH 2005, Gliwice, pp. 47–48 (2005)
18. Heppner, F., Grenander, U.: A stochastic nonlinear model for coordinated bird flocks. In: Krasner, S. (ed.) The Ubiquity of Chaos, AAAS Publications, Washington, DC (1990)
19. Hwang, S.-F., Wu, J.-C., He, R.S.: Identification of effective elastic constants of composite plates based on a hybrid genetic algorithm. Composite Structures 90, 217–224 (2009)
20. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimisation. In: Proceedings of IEEE Int. Conf. on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)
21. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann (2001)
22. Mrozek, D., Małysiak-Mrozek, B.: An Improved Method for Protein Similarity Searching by Alignment of Fuzzy Energy Signatures. International Journal of Computational Intelligence Systems 4(1), 75–88 (2011)
23. Poteralski, A., Szczepanik, M., Dziatkiewicz, G., Kuś, W., Burczyński, T.: Immune identification of piezoelectric material constants using BEM. In: Inverse Problems in Science and Engineering, vol. 19(1). Taylor & Francis
24. Pan, E.: A BEM analysis of fracture mechanics in 2D anisotropic piezoelectric solids. Engineering Analysis with Boundary Elements 23, 67–76 (1999)
25. Ptak, M., Ptak, W.: Basics of Immunology. Jagiellonian University Press, Cracow (2000)
26. Reynolds, C.W.: Flocks, herds, and schools, A distributed behavioral model. Computer Graphics 21, 25–34 (1987)
27. Silva, M.F.T., Borges, L.M.S.A., Rochinha, F.A., de Carvalho, L.A.V.: A genetic algorithm applied to composite elastic parameters identification. In: IPSE, vol. 12, pp. 17–28 (2004)
28. Tan, K.C., Goh, C.K., Mamun, A.A., Ei, E.Z.: An evolutionary artificial immune system for multi-objective optimization. European Journal of Operational Research, 371–392 (2008)
29. Warwick, K., Kang, Y.-H., Mitchell, R.J.: Genetic least squares for system identification. Soft Computing 3, 200–205 (1999)
30. Zieniuk, E., Gabrel, W.: Genetic algorithms based on a new system of integral equations in identification of material constants for anisotropic media. Mechanics of Composite Materials 37, 217–222 (2001)
31. Wierzchoń, S.T.: Artificial Immune Systems: Theory and Applications. EXIT Press (2001)
32. Zilong, G., Sun'an, W., Jian, Z.: A novel immune evolutionary algorithm incorporating chaos optimization. Pattern Recognition Letters 27, 2–8 (2006)

Universal Intelligence, Creativity, and Trust in Emerging Global Expert Systems

Andrzej M.J. Skulimowski[1,2]

[1] AGH University of Science and Technology,
Chair of Automatic Control and Biomedical Engineering,
Decision Science Laboratory, 30-050 Krakow
[2] International Centre for Decision Sciences and Forecasting,
Progress & Business Foundation, 30-041 Kraków, Poland
ams@agh.edu.pl

Abstract. This paper presents a hypothesis together with evidence related to the use of global knowledge as a holistic expert system. By global expert system (GES) we mean all knowledge sources, bases, repositories, and processing units, regardless of whether they are human, artificial, animal, or hybrid, such that the relation "*able to transfer knowledge on immediate demand*" forms a connected graph over the elements of the system. A key requirement is that problem solving using GES is an anytime process with respect to the number of information sources taken into account. We conjecture that due to the high and ever-growing level of interconnection of knowledge units, a universal intelligence emerges, which under specific conditions can outperform the intelligence and creativity of any of its individual elements, including humans. It will be shown that this is possible only if an appropriate level of credibility can be assigned to each element of the system, which ensures that users trust the responses. We will design a hybrid supervised-reinforced learning scheme that makes it possible to achieve a satisfactory level of trust in GES query responses. Query processing will apply knowledge fusion methods such as combinations of recommendations and forecasts.

Keywords: Global expert systems, big data, universal intelligence, trust management, semi-supervised learning, foresight.

1 Introduction

This paper presents several problems related to the emergence of global knowledge regarded and used as a holistic global expert system. By a global expert system (GES) we mean all knowledge sources, sensors, bases, repositories and processing units, regardless of whether they are human, artificial, animal, or hybrid, provided that they are mutually connected and endowed with an information management system with the usual expert system functionalities. Specifically, in a GES the relation "able to transfer knowledge on immediate demand" forms a connected graph over the elements of the global knowledge system.

The interest in GES results mainly from the emergence of global information exchange networks powered by search engines and autonomous network crawlers.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 582–592, 2013.

Three general development trends related to GES have been identified within the research project [18]:

- a growing level of integration of heterogeneous information sources
- a growing number of interconnected knowledge units
- an increase in the amount of information and sophistication of information processing within individual units.

These are supplemented by qualitative trends regarding the degree of refinement of the information stored and processed in each unit according to the well-known scheme: information→knowledge→wisdom. Other trends touch upon the structure of this information. For instance, the percentage of all data stored on the web and indexed by the Google search engine rose from 1% in January 2007 to 6% in January 2010 and exceeded 10% in January 2012. At the same time, the estimated amount of information on the web rose to 800 exabytes and the number of web sites exceeded 560 million (cf. e.g. [21] and further links there). When the tools offered by search engines become increasingly sophisticated, this system of interconnected web sites becomes a real GES endowed additionally with a variety of analytic methods.

The second source of inspiration for the underlying research comes from studies on collective intelligence [19],[20], specifically from a group of knowledge elicitation methods described by the common term real-time Delphi [6], where experts reply to structured questions, and the knowledge thus gathered is verified, fused, further processed, and merged with knowledge from other sources. This is the typical approach used in complex decision making and foresight studies. We will explore the idea that any kind of information source in a GES can be dealt with in a similar manner to experts in a Delphi, while Delphi-specific mechanisms that limit the number of knowledge elicitation steps can be adopted and used to define the stopping criteria for information processing in GES.

A fundamental assumption is that problem solving using a GES should be an anytime process [3], monotonic with respect to the number of information sources taken into account and to the complexity of analytic procedures. In Sec.3 we will formulate the hypothesis that due to the high and ever-growing level of interconnection of knowledge sources, sensors, processing units, knowledge bases and repositories, a universal intelligence emerges, which under certain conditions can outperform the intelligence and creativity of any of its individual elements, including humans. It will be shown that this is possible only if an appropriate level of credibility can be assigned to each element of the system in order to ensure a certain level of trust in the system outputs on part of its users.

Trust and credibility management turns out to be the first main issue that can hinder or allow the use of heterogeneous knowledge repositories and their interconnected systems as a GES. A basic principle related to trust management, presented below, is the distinction between trust in an individual knowledge source and the credibility of information received from it. This corresponds to a well-known fact that a distrusted source of information that has a high probability of providing false responses can nevertheless be a useful source of information provided that a lower probability is assigned to the actual response of this unit than to its complementary statement.

The second relevant issue that needs to be considered when designing a GES is the choice of a search-and-survey strategy to process a query reviewing a very large

number of feasible information sources. The survey planning approach, presented e.g. in [14], cannot be used in a dynamically changing environment with a very large number of potential knowledge sources, out of which only a quotient is explicitly known *ex-ante*. Also, classical precision-and-reward assessment of responses to the query will fail for a number of reasons. In particular, the user will not be able to assess the results on his/her own and will be forced to delegate the judgment regarding the quality of the reply and corresponding decisions to autonomous agents. A heuristic search-and-survey procedure can be designed making use of the *creative decision process* notion [15], where the user defines an initial subset of information sources according to some criteria, assigns them trust coefficients and activates the procedure that runs recursively at each information source, transforming them to autonomous agents with similar capabilities as the user. This allows the search to be pursued by these agents in a deeper web autonomously and simultaneously, up to a prescribed stack level. Assuming that at each level N information sources on average can be specified for the next-step search, and that the procedure prevents retrieving information from any site two or more times in one run, we obtain $N(N^K-1)/(N-1)$ information sources that can be surveyed in K steps, a sufficiently large quantity when compared with the number of websites ($\sim 10^9$) in existence nowadays.

In Sec. 2 we will design a hybrid supervised-reinforced learning scheme that can assist in achieving a satisfactory level of trust in GES responses. We will also apply knowledge fusion methods [16], including the anticipatory networks [17], Ashton [1], and Hogarth [7] approaches to combine qualitative recommendations and forecast combination methods preserving different optimality criteria [2],[4]. In Sec.3 we will show the synergy of the above fusion and trust models in an emerging GES. Sec.4. presents a real-life example related to trust management in an IT foresight exercise.

2 Trust Management in a GES

The main purpose of using a trust and credibility management model in a GES can be presented as follows. We have claimed that a query response from a GES endowed with a suitably designed trust management functionality can outperform a response provided by the same GES acting on a subcomponent of its knowledge base in terms of expected *ex-ante* return information value. This is another formulation of the anytime property with respect to knowledge resources used. It prevents the well-known threat of spoiling credible information by merging it with additional unverified data. The value of the return information is to be defined as a performance index related to the benefits from the further use of this information [14] by the recipient of the response. If the response is quantitative, its value is usually a monotonic function of the accuracy that can be measured *ex-ante* as the standard deviation from the expected value.

The desired properties of the GES are achieved due to the action of intelligent algorithms capable of fusing the knowledge gathered from individual sources, taking into account different competence levels and different degrees of credibility among various types of information sources. The instances of the latter, when regarded as elements of a GES, will be termed here *knowledge units* (KU). We will assume that an agent that generates the query and receives the response is a knowledge unit as well, i.e. the refined knowledge gathered by this agent can be retrieved later by other

KUs. Such an agent will be termed an *active knowledge unit* (AKU), as long as the knowledge management system of the GES is activated by this agent.

The credibility of each KU is modelled by a vector whose coordinates correspond to each subject area in a query that is independent from the other areas. This vector describes the ability to provide correct answers in the areas of interest. In addition, trust coefficients describe the KU's ability to provide genuine answers, based on full knowledge resources. Observe that these assumptions differ from the most widespread Dempster-Shafer-based trust and confidence models [5],[8],[11] as well as from the rough-sets-based credibility model [13]. This is justified by the way the trust and credibility coefficients are updated in our model using the reinforcement learning scheme that better fits an update of a single coefficient of a KU at one time than an update of three or four belief/disbelief/uncertainty measures. Another justification of using single trust coefficients comes from the way they are further used to combine recommendations and forecasts, where the aggregate value is a linear combination of individual statements. Nevertheless, it is expected that our approaches would lead to similar results as the other two mentioned above, as the mutual monotonicity of coefficients used in these approaches is preserved by iterative update rules.

Once substantial credibility and trustworthiness has been described numerically, an aggregation function may be used to fuse the information from different KUs, using the maximum likelihood principle, similar to combining recommendations or forecasts [1],[2]. If the estimated distribution has several maxima or if its variance exceeds a certain threshold, then the survey outputs are clustered and a prescribed number of scenarios is produced using the Bates and Granger [2] method to combine forecasts based on the maximum likelihood principle [6]. The credibility vectors and trust coefficients will be updated recursively, taking into account the updates caused by one's responses in other opinions. The learning scheme applies supervisory or weak supervisory learning models that can be implemented as a hybrid neural network with generalised neurons that model KUs. This model can be used to manage the credibility of all KUs and heterogeneous sources of information such as experts, experts systems, sensors, web databases, books, articles etc. within a knowledge building process.

The neural supervised learning model will be applied to update credibility vectors assigned to different sources of information, while moderated self-assessment is used to define the initial values. We will distinguish between trust in the information source and credibility of a particular instance of information produced by this source. The latter may be low even though we trust the source of the information which claims to be unsure of the query issue. Response errors detected *ex-post* by a supervisor will be quantified for qualitative replies as well and transformed into increments (or decrements) of the credibility vector. Such quantification requires a definition of a proximity measure over all potential qualitative responses. This results in a combined trust-credibility model which can be applied in a GES emerging from a large number of heterogeneous interconnected databases.

The fundamental assumption of credibility management for a GES is that a coefficient $c_{ij} \in [-1,1]$, termed credibility of the i-th KU in the j-th subject area, can be assigned to each KU that belongs to the GES. For a fixed i, the coefficients c_{ij} form the competence vector C_i corresponding to the i-th KU. Then C_i are used as

coefficients of the aggregation functions that merge the responses to the query provided by the individual KU. The values of C_i that yield minimal error are sought.

Initial values of C_i can be defined in various ways, based on the initial assessments by the user, on the self-assessments of the expert KUs, as well as possibly resulting from earlier searches. If the supervised learning scheme can be applied to credibility management and to optimize the trustworthiness of the query responses at an AKU then the information processing by a GES can be organized analogously to a two-round Delphi exercise [10], as follows:

Procedure 1

Step 1. The criteria to select the KU are defined by the AKU as well as criteria to transform a KU to the next-level AKU.
Initial credibility coefficients are assigned to all KUs in the subject areas of the present and subsequent queries.

Step 2. KUs respond to the queries, either qualitatively (multiple choice replies selected from the user-defined list) or quantitatively (numbers, functions etc.), provide justifications, and feed them to the AKU.

Step 3. The AKU, possibly assisted by experts, reviews the responses and modifies the KU trust coefficients based on the knowledge supplied by all experts.

Step 4. The AKU assisted by an expert panel clusters the 2nd round replies and – based on this AKU's choice criteria - selects either
- a consensus, which ends the procedure by skipping to Step 6,
or
- accepts a dissensus by defining a number of plausible reply scenarios, the procedure goes to Step 6, or
- defines further AKUs from among KUs to pursue the search; the present process becomes idle until the next-level search results are obtained.

Step 5. The next-level AKUs are activated unless the *K*-th search level or a time constraint is reached, in the latter case go to Step 6.

Step 6. The final results are presented to the higher-level AKU (for K>1) or to the user (for K=1).

The experts in Steps 3 and 4 can be either human experts assisting the user or autonomous agents that store the results of earlier searches by different users and are able to modify the trust coefficients and clustering criteria. At a search level greater than *1* they may also define the KU's selection criteria for next-level search. The autonomous agents can be provided as the service of a web company, for instance by the operator of a search engine or independent companies. They will be termed *knowledge recommenders*.

So far we have made no assumption about the structure of the query. Now let us assume that the AKU processes a complex query consisting of *J* questions that are replied by *M* different KUs. The responses can be either numerical or may belong to a finite set of attributes. The learning principle used to update the KU credibility vectors within the 2nd round of query processing at an AKU can now be formulated as

$$\hat{c}_{ij} := s(c_{ij} + \sum_{k=1}^{M} c_{kj}\delta_{kj}(r_{ij}, r_{kj}, \hat{r}_{kj})) \tag{1}$$

where c_{ij} is the credibility of the i-th KU in the j-th area, r_{ij} is the first round reply of the i-th KU to the j-th question, $\hat{r}_{kj}$ is the k-th KU reply to the j-th question in the 2^{nd} round, updated after knowing all 1^{st} round replies and the supplementary justification material (if any). Further, δ_{kj} are functions transforming the value of the change of the k-th KU response to the j-th question influenced by another (e.g. i-th) KU response in the 1^{st} round. The threshold function that maps IR monotonically in $[-1,1]$ may be defined as

$$s(x)=x \text{ iff } x\in[-1,1] \text{ else if } |x|>1 \text{ then } s(x)=sgn(x). \tag{2}$$

Alternatively, a sigmoid transformation may be used. We assume that the initial values of c_{ij} are always non-negative. Trust coefficients c_{0i} are updated similarly to (1) based on contradictory statements in the responses and the deviations between the self-assessment of credibility in an area and the quality of justifications assessed by 2^{nd} round experts. They may be used as additional adjustment coefficients in (1).

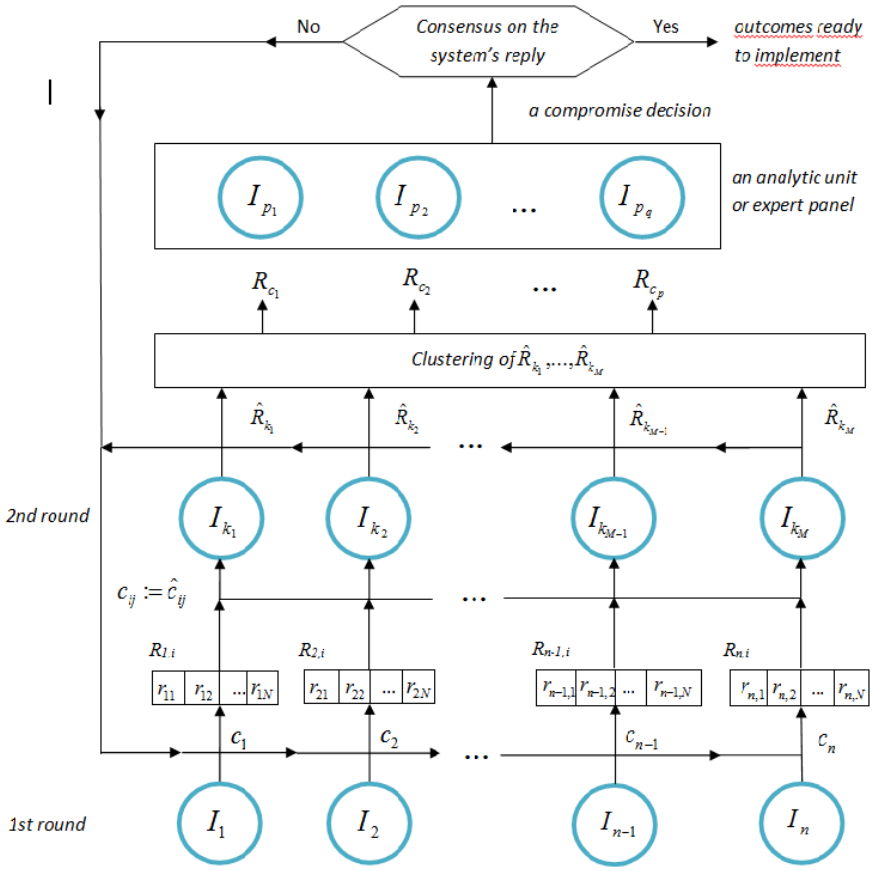


Fig. 1. A scheme of supervised learning applied to the trust coefficients $c_1,\ldots,c_n$ of information sources $I_1,\ldots,I_n$. in a GES

The δ_{kj} are to be selected in such a manner so that the optimization problem

$$\sum_{p=1}^{P}\sum_{i=1}^{n(p)} (r_{cpj}(\hat{c}_{1j},...,\hat{c}_{nj}) - r_{ij})^2 \to \min, \quad for\ j = 1,...,N \tag{3}$$

where r_{cpj} are p-th scenario values of the j-th reply after clustering, $p=1,...P$, and $n(p)$ is the cardinality of the p-th cluster. The inner summation is performed over the responses that were assigned to the p-th cluster.

Let us note that (2) is in fact a bi-level multicriteria optimization problem, where the optimality of c_{ij} in (2) depends on a prior finding of the optimal δ_{kj} in (1). If the number of scenarios P is not a priori given then P becomes additional variable to be optimized in (2). If $P=1$ then (2) reduces to a consensus finding problem. The above procedure is shown in Fig.1.

3 An Application of the Fusion of Expert Information and Quantitative Trend Models

The knowledge processing scheme applying the credibility and trust of experts interacting in a structured way and other knowledge units selected from a plethora of potentially useful information sources turned out useful in designing the Delphi exercise in a foresight project on the Information Society technologies carried out recently [13]. Many other recent foresight exercises (cf. e.g. [6]) use computer-assisted Delphi questionnaire research as one of their knowledge acquisition tools. In [18] the same trust-management scheme has been applied to define the information retrieval strategy from the open web and from the bibliographic databases. Finally, the scheme presented in Sec.2 has been used in information fusion algorithms. Different procedures to combine knowledge and forecasts and to generate recommendations to the decision-maker have been elaborated from the knowledge gathered.

One of the main problems that arise when analysing Delphi research outputs is the diversified trustworthiness of individual respondents who may possess different qualifications, expertise, and access to information and modelling tools. In addition, experts' responses may be burdened by a tendency to present views that coincide with individual gains rather than provide an objective picture. This problem is especially relevant where the research covers a large multi- or interdisciplinary area, when it is difficult to find an expert in all areas simultaneously. A similar problem arises when there is a need to involve different groups of practitioners, students and stakeholders, which may provide information with varying degrees of trustworthiness in specific or all areas, or if the exercise uses crowdsourcing.

The credibility of each expert has been modelled by a vector whose coordinates correspond to each of the following areas in the Delphi questionnaires:

- the economic and social aspects of Information Society development
- e-commerce, e-government, and e-business
- decision support systems and recommenders
- diagnostic and embedded expert systems
- computer vision

- robotics and autonomous intelligent systems
- neurocognitive systems
- molecular computing
- quantum computing.

This vector describes the ability to provide correct answers in the areas covered by the survey. In addition, trust coefficients describe the participants' ability to provide genuine answers. To apply Procedure 1 to analyse the outputs of an expert Delphi consisting of responding to structured questionnaires on IT development trends, we have to define the initial trust coefficients as presented in Tab.1 below.

Table 1. The assignment of initial competence scores to experts (self-assessment)

Verbal Self-Assessment of Competence in a Survey Area	Verbal Self-Assessment of Practical Skills in an Area	Numerical value	Normalised score
I am not interested in this issue (the question may be skipped)	I have no experience with these methods/topics	1	0/0,05 (0 if no reply)
I only have some general knowledge in this area (but I shall try to fill in the questionnaire)	I know some articles reporting the practical experience in this area or I use these methods sporadically	2	0,25
I feel confident with some of the issues in this area	I have been regularly using some of these methods / applications	3	0,5
I am confident with all the relevant issues in this area	I am an experienced user of most of these methods and tools	4	0,8
I am a specialist in this area	I am a specialist-practitioner	5	1

The numerical coefficients input to Procedure 1 have been calculated as products of normalised scores and initial trust coefficients, for each subject area separately. Once substantial credibility and trustworthiness have been described numerically, the expected *ex-ante* error of the combined output can be calculated using e.g. the approaches presented in [1] or [4]. The ultimate goal of selecting the most appropriate trust and credibility coefficients was to minimise this error. If the estimated distribution has several maxima or if its variance exceeds a certain threshold, then the survey outputs are clustered and a prescribed number of scenarios is produced based on the maximum likelihood principle. During the analysis of results, the credibility vectors and trust coefficients are updated recursively, taking into account the updates caused by the assessments of responses of the Delphi participants by the core expert group. The opinions of other respondents of the same exercise can also be taken into account as well as the contradictions in the responses that can be detected automatically. The learning scheme applies neural network supervisory learning models.

Further analysis may eventually lead to defining bibliographic sources to complement the missing or incomplete expert responses, using the recursive k-th to $(k+1)$-information source level transition scheme presented in Sec.2. Multi-round Delphi studies can be dealt with within a similar framework. The combination of the *ex-ante* error analysis with the usual *ex-post* approaches could convert the semi-supervised trust learning scheme to a supervised scheme as soon as the verification of foresight

results derived from Delphi are possible, usually only after a long time lapse. A further discussion of this approach, and its results in IST foresight, is given in [18].

4 Discussion and Conclusions

The research on the credibility of experts in a Delphi exercise has been first extended to deal with credibility and trust in group model building systems aimed at elaborating complex socio-economic evolution models [16]. Then, the common expert reputation management system has been used for several analytic engines to optimize the choice of experts to taking part in particular exercises from the point of view of simultaneously maximising the probability of judgments and the statistical trustworthiness of outputs [19]. Further application of decision-making creativity in a GES framework, according to the notion presented in [15], will allow us to enhance the autonomy of the knowledge gathering process in Delphi exercises by the automatic search for web, patent, and bibliographic information, as well as to request opinions of additional experts.

Creativity in a GES has at least a twofold meaning: it can be artificial creativity (delegating complicated search strategy selection tasks to autonomous agents) as well as creativity of users aided by intelligent agents. Furthermore, it can be expected that the opinion dynamics and their distribution within the same group of experts can be forecasted using the state-space model and Kalman filtering as proposed in [11].

An important feature of the recursive search procedure and judgment fusion algorithms proposed above is its universality. Specifically, the GES framework will allows users to manage trust, confidence, and credibility of heterogeneous sources of information, such as experts, expert systems, web databases, books and articles etc., in a uniform way. Even unattainable information sources (e.g. absent experts) can be assigned credibility vectors and trust coefficients, using the Kalman filter in a similar way as proposed above. Queries related to the future can be modelled and replied to within an anticipatory networks framework [17] coupled with forecasting models such as the Kalman filter or VARIMA [12].

Another challenging problem is the evolution of the system of semi-autonomous agents that perform a knowledge search in a future heterogeneous web. They can be termed semi-autonomous because they get task from higher-level agents, but the way the tasks are fulfilled can be chosen autonomously. It can be conjectured that such agents show a self-organisation behaviour similarly to the interacting brain regions considered in [9]. If a brain-like hierarchy can be followed by artificial structures of very different origin and nature, driven only by the similarity of evolutionary mechanisms, the consequences might be amazing: a spontaneous emergence of a super brain in large artificial networks, without any biological limitations, and potentially able to produce brain-like states.

Acknowledgement. The results presented in this paper have been obtained during the research project “Scenarios and Development Trends of Selected Information Society Technologies until 2025” financed by the ERDF within the Innovative Economy Operational Program 2006-2013, Contract No. WND-POIG.01.01.01-00-021/09.

References

1. Ashton, A.H., Ashton, R.H.: Aggregating subjective forecasts: Some empirical results. Management Science 31, 1499–1508 (1985)
2. Bates, I.J.M., Granger, C.W.J.: The combination of forecasts. Operational Research Quarterly 20, 451–468 (1969)
3. Dean, T.L., Boddy, M.: An analysis of time-dependent planning. In: Proceedings of the Seventh National Conference on Artificial Intelligence, AAAI 1998, St. Paul, MN, pp. 49–54 (1988)
4. Elliott, G., Timmermann, A.: Optimal forecast combinations under general loss functions and forecast error distributions. Journal of Econometrics 122, 47–79 (2004)
5. Gligor, V., Wing, J.M.: Towards a Theory of Trust in Networks of Humans and Computers. In: Christianson, B., Crispo, B., Malcolm, J., Stajano, F. (eds.) Security Protocols 2011. LNCS, vol. 7114, pp. 223–242. Springer, Heidelberg (2011)
6. Gnatzy, T., Warth, J., von der Gracht, H., Darkow, I.-L.: Validating an innovative real-time Delphi approach - A methodological comparison between real-time and conventional Delphi studies. Technological Forecasting & Social Change 78, 1681–1694 (2011)
7. Hogarth, R.: A note on aggregating opinions. Organizational Behavior and Human Performance 21, 40–46 (1978)
8. Jøsang, A.: A Logic for Uncertain Probabilities. Int. J. Uncertainty, Fuzziness Knowl.-Based Syst. 9(3), 279–311 (2001)
9. Joyce, K.E., Laurienti, P.J., Hayasaka, S.: Complexity in a brain-inspired agent-based model. Neural Networks 33, 275–290 (2012)
10. Linstone, H.A., Turoff, M. (eds.): The Delphi Method. Techniques and Applications. Electronic version © Harold A. Linstone, Murray Turoff, p. 616 (2002)
11. Moyano, F., Fernandez-Gago, C., Lopez, J.: A Conceptual Framework for Trust Models. In: Fischer-Hübner, S., Katsikas, S., Quirchmayr, G. (eds.) TrustBus 2012. LNCS, vol. 7449, pp. 93–104. Springer, Heidelberg (2012)
12. Öller, L.E.: Macroeconomic forecasting with a vector arima model: A case study of the finnish economy. International Journal of Forecasting 1(2), 143–150 (1985)
13. Podraza, R.: Credibility Coefficients in Hybrid Artificial Intelligence Systems. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 187–194. Springer, Heidelberg (2009)
14. Skulimowski, A.M.J.: Optimal strategies for quantitative data retrieval in distributed database systems. In: Proceedings of the Second International Conference on Intelligent Systems Engineering, Hamburg, September 5-9, pp. 389–394. IEE Conference Publication Nr 395, IEE, London (1994)
15. Skulimowski, A.M.J.: Freedom of Choice and Creativity in Multicriteria Decision Making. In: Theeramunkong, T., Kunifuji, S., Sornlertlamvanich, V., Nattee, C. (eds.) KICSS 2010. LNCS, vol. 6746, pp. 190–203. Springer, Heidelberg (2011)
16. Skulimowski, A.M.J.: Discovering Complex System Dynamics with Intelligent Data Retrieval Tools. In: Zhang, Y., Zhou, Z.-H., Zhang, C., Li, Y. (eds.) IScIDE 2011. LNCS, vol. 7202, pp. 614–626. Springer, Heidelberg (2012)
17. Skulimowski, A.M.J.: Anticipatory Network Models of Multicriteria Decision-Making Processes. Int. J. Systems Sci. 44 (2012), doi:10.1080/00207721.2012.670308

18. Skulimowski, A.M.J. (ed.): Scenarios and Development Trends of Selected Information Society Technologies until 2025. Final Report. Progress & Business Publishers, Kraków, Progress & Business Foundation (2013), http://www.ict.foresight.pl
19. Tapscott, D., Williams, A.D.: Wikinomics: How Mass Collaboration Changes Everything, 3rd ed., Portfolio Trade (2010)
20. Tovey, M. (ed.): Collective Intelligence: Creating a Prosperous World at Peace. Earth Intelligence Network, Oakton, VA (2008)
21. http://en.wikipedia.org/wiki/Exabyte (accessed October 15, 2012)

A Swarm Intelligence Approach to Flexible Job-Shop Scheduling Problem with No-Wait Constraint in Remanufacturing

Shyam Sundar[1], P.N. Suganthan[1], and T.J. Chua[2]

[1] School of Electrical & Electronic Engineering,
Nanyang Technological University Singapore,
639798, Singapore
{ssundar,epnsugan}@ntu.edu.sg
[2] Singapore Institute of Manufacturing Technology,
71, Nanyang Drive Singapore,
638075, Singapore
tjchua@simtech.a-star.edu.sg

Abstract. This paper addresses a flexible job-shop scheduling problem with no-wait constraint (FJSPNW) which combines features of two well-known combinatorial optimization problems – flexible job-shop scheduling problem and no-wait job-shop scheduling problem. To solve FJSPNW with the objective of minimizing the makespan, an artificial bee colony (ABC) algorithm is proposed. This problem finds application in remanufacturing scheduling systems. ABC algorithm is a recently developed swarm intelligence technique based on intelligent foraging behavior of honey bee swarm. Since its inception, it has shown promising performance for the solution of numerous hard optimization problems. Numerical experiments have been performed on a set of standard benchmark instances in order to demonstrate the effectiveness of ABC algorithm.

Keywords: Swarm Intelligence, Artificial Bee Colony Algorithm, No-wait, Flexible Job-shop, Scheduling.

1 Introduction

In recent years, various scheduling problems have been studied extensively due to their theoretical and practical applications in the planning and manufacturing systems. Job-shop scheduling problem (JSP) is one among them. JSP is a well-known $\mathcal{NP}$-Hard problem [1] combinatorial optimization problem which seeks to schedule n jobs on m machines with some criteria. Each job consists of a predetermined sequence of non-preemptable operations and each operation needs a machine to process. Flexible job-shop scheduling problem (FJSP) [2–5] and no-wait job-shop scheduling problem (NWJSP) [6–10] are also two well-known hard combinatorial optimization problems which are generalization of JSP. FJSP allows an operation to be processed by any machine out of a set of machines, whereas NWJSP doesnot allow waiting time between operations associated with

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 593–602, 2013.

a job, i.e., all operations associated with a job must be processed continuously one by one without any interruption.

In this paper, we consider a generalized problem called flexible job-shop scheduling problem with no-wait constraint (FJSPNW) which combines both features of FJSP and NWJSP. This problem finds application in remanufacturing scheduling systems. In FJSPNW, a set of n jobs $J_1, J_2, \ldots, J_n$ and a set of m machines $M_1, M_2, \ldots, M_m$ are given. Each job J_i has a predetermined sequence of non-preemptable N_i operations $\{O_{i1}, O_{i2}, \ldots, O_{iN_i}\}$. Each operation O_{ij} can be processed by any machine from a predetermined set of machines. P_{ijk} is the processing time of operation O_{ij} on machine k. All jobs and machines are available at time 0. At a given time, each machine can process at most one operation. No waiting time is allowed between operations associated with a job. In other words, all operations associated with a job must be processed continuously one by one without any interruption. Thus, FJSPNW aims to assign each operation to an appropriate machine and to find out a sequence of jobs with no-wait constraint in such a way that some objective is met. This paper addresses FJSPNW with the objective of minimizing the makespan.

To the best of our knowledge, there is no reported work for FJSPNW in the literature. Since FJSPNW is a generalization of FJSP and JSPNW, therefore, FJSPNW is more complex problem than FJSP and JSPNW. In such a condition, metaheuristic techniques can be effective techniques in finding high quality solutions in a reasonable time. It has been seen in recent years that swarm intelligence techniques characterized by the collective behavior and self-organized behavior of swarms have been applied effectively to solve numerous hard optimization problems. Artificial bee colony (ABC) algorithm inspired by intelligent foraging behavior of honey bee swarm [11] is one among them. Since its inception, it has shown promising performance and competitiveness with other state-of-the-art metaheuristic techniques. Initially, it was designed for optimization problems in continuous domain. Later, it was further carried out for discrete optimization problems [12–15]. A detailed survey of the applications of ABC algorithm can be studied in [16]. This paper presents an ABC algorithm for the solution of FJSPNW and the performance of ABC algorithm is tested on a set of standard benchmark instances.

The rest of this paper is organized as follows: Section 2 outlines a brief introduction to ABC algorithm. Section 3 presents ABC algorithm for FJSPNW which will be referred to as ABC_FJSPNW. Section 4 demonstrates computational results, whereas Section 5 concludes the proposed work.

2 ABC Algorithm

ABC algorithm developed by Karaboga [11] is a swarm intelligence technique based on intelligent foraging behavior of honey bee swarm. The colony of real bees is classified into three groups – Scout, employed and onlooker bees. Scout bees are those bees that are searching new food sources in the vicinity of the hive. When the search of discovering a food source is successful, scout bee becomes

employed bee. Employed bees are those bees that are currently exploiting food sources. After exploiting, they carry loads of nectar from the food sources to the hive and participate in communicating information about their food sources to the onlooker bees by means of dancing. The nature and duration of the dance performed by the dancing bee depend on the richness of the food source. Onlooker bees are those bees that are waiting for the employed bees. After arrival of employed bees, onlooker bees watch numerous dances of employed bees and then, they select a food source with a probability which is directly proportional to the nectar content of that food source. It helps in attracting more and more number of onlooker bees towards a good quality of a food source. As soon as, an onlooker bee selects a food source, it becomes an employed bee. If a food source is exploited completely, then this food source is abandoned by those employed bees that associate with this food source and all these employed bees become either scouts or onlookers.

In ABC algorithm [11], artificial bees are also classified into three groups – scout, employed and onlooker bees. Each food source represents a feasible solution to the problem under consideration, whereas its nectar content represents the fitness of the solution. Since in ABC algorithm, each food source is associated with an employed bee. So, the number of food sources is equal to the number of employed bees. ABC algorithm begins with initializing a fixed number of solutions randomly. Then, a search process that includes two following phases is carried out repeatedly until the termination criterion is satisfied.

1. *Employed Bee Phase*: Each employed bee determines a new food source (solution) in the neighborhood of its associated food source. If the new food source is better than its currently associated solution based on fitness, then it moves to the new neighboring food source, otherwise it continues with the old one. If a food source doesnot improve for *limit* number of trials (iterations), then it is presumed that this solution is exploited completely. In such a situation, employed bee that associates with this food source abandons its food source and becomes a scout bee. As soon as, this scout bee generates a new solution (food source) randomly, it again becomes an employed bee.
2. *Onlooker Bee Phase*: All employed bees participate in communicating information about their fitnesses with onlooker bees. With the help of probability based selection method which biases towards the good quality of solutions (food sources) over the bad ones, each onlooker bee selects a solution. A good solution will be selected by more and more number of onlooker bees. All onlooker bees, similar to employed bee phase, also determine new food sources in the neighborhood of its selected food source. After that, among all new neighboring food sources determined by onlookers that associate with a particular food source k and food source k itself, the best solution will be assigned to the position of k^{th} food source. This phase is completed when the new positions of all food sources are determined.

3 ABC_FJSPNW

Main characteristics of ABC_FJSPNW are described as follows:

3.1 Solution Encoding

Since FJSPNW is characterized by two things – sequence of jobs with no-wait constraint and assignment of operations to machines, therefore, in order to encode (represent) a solution, a list called *job_list* is used to maintain the sequence of jobs. In addition, for each job i in *job_list*, another list called *machine_list* of all selected machines used for processing their operations $\in i$ is used. It is to be noted that each index of *machine_list* represents the operation number of the corresponding job.

3.2 Initialization

In order to maintain a balance between diversity and quality in the population, an iterative procedure which consists of random and greedy strategies is applied to generate each initial solution S of the population. Initially, S is an empty solution and all jobs are added to a set called U. A job i is selected uniformly at random from U. This selected job i is added to the first empty position of *job_list* of S. Hereafter, in order to process each operation j of this selected job i, i.e., O_{ij} without violating no-wait constraint, iteratively a machine k is selected for O_{ij} from its predefined set. Selection of a machine k for O_{ij} from its predefined set is done either randomly or greedily. With probability P_r, a machine is selected uniformly at random from its predefined set, otherwise a machine having minimum processing time is selected with probability 1 - P_r. P_r is an empirical parameter. This machine k for O_{ij} is added to j^{th} index of *machine_list* of the selected job i. This whole iterative procedure continues until assignment of all operations $\in i$ to their machines is completed. Selected job i is deleted from U. This whole iterative procedure continues until U becomes empty.

To ensure the feasibility of a solution after the generation of a solution, a solution decoding procedure is applied to decode the solution which is explained in the next subsection.

3.3 Solution Decoding

Since FJSPNW is a generalization of NWJSP, therefore, solution decoding procedure for NWJSP can also be applied for FJSPNW. The procedure for solution decoding [18] is described as follows:

Step 1. Initially, each machine is allotted with zero to infinity idle time interval.

Step 2. Select a job in the sequence of the schedule (solution), then find all matching among idle time intervals of machines and the processing times of all operations of the current job without violating no-wait constraint.

Step 3. Update idle time interval of each machine.
Step 4. If all jobs are chosen, then the procedure stops. Otherwise goto Step 2.

It is to be noted according to [18] that if the processing time of any operation of the current job doesnot succeed to match with its corresponding machine's idle time interval, then that operation is delayed until it finds a corresponding idle time interval, and simultaneously every other operation of the same job is also delayed by the same amount and also a re-check is done so that the processing time of every operation of the current job matches with its corresponding machine's idle time interval without violating no-wait constraint. Basically, this decoding procedure delays the start time of each job of the schedule so that the processing times of its all operations fall in their corresponding machines' idle time interval and it also doesnot violate no-wait constraint.

Each candidate solution is uniquely associated with an employed bee. The fitness (makespan) of each solution is calculated.

3.4 Probability of a Selecting a Food Source

Each onlooker bee selects a food source (solution) with the help of binary tournament selection method which is a probability based selection method. In this method, two different food sources are selected uniformly at random from the population. With probability P_{bt}, best of them is selected, otherwise worse one is selected. P_{bt} is an empirical parameter.

3.5 Determination of a Neighboring Food Source

In order to find a high quality solution (food source), the problem structure must be exploited as much as possible. Since the problem structure of FJSPNW is based on the sequence of jobs with no-wait constraint and assignment of operations to machines, therefore, two different methods – multi-point insert method [17] and perturbation method – are used in a mutually exclusive way for determining a solution Y in the neighborhood of a solution X. Multi-point insert method uses the concept of utilizing solution components from another solution [12][17]. It is also based on this assumption that if a component is positioned at right place in a good solution (schedule), then there is high possibility that this component should be exactly at the same position or vicinity to the same position in many other good solutions. Whereas, perturbation method perturbs the sequence of jobs and assignment of operations to machines in order to avoid the solution to be trapped in a local optima. With probability P_{mpi}, multi-point insert method is applied, otherwise perturbation method is applied with probability 1-P_{mpi}. P_{mpi} is an empirical parameter.

1. *Multi-point Insert Method*: In this method, an another solution Z which is different from X is selected randomly from the population. Initially, Y is an empty solution. After that, *nrp* different positions are selected randomly

Algorithm 1. Pseudo-code of ABC_FJSPNW

```
Initialize EP solutions e_1, e_2, ..., e_EP;
BEST ← Best solution ∈ {e_1, e_2, ..., e_EP};
while Termination criteria is not satisfied do
    for l ← 1 to EP do
        e_nbr ← Nbring_Sol(e_l);
        if e_nbr is better than e_l then
            e_l ← e_nbr;
        else if e_l has not improved for a predetermined number of iterations,
        i.e., limit iterations then
            Scout bee;
        if e_l is better than BEST then
            BEST ← e_l;
    for l ← 1 to OP do
        I_l ← Selection(e_1, e_2, ..., e_EP);
        o_l ← Nbring_Sol(e_{I_l});
        if o_l is better than BEST then
            BEST ← o_l;
    for l ← 1 to OP do
        if o_l is better than e_{I_l} then
            e_{I_l} ← o_l;
```

from *job_list* of Z. All jobs corresponding to selected positions as well as their associated machines in their *machine_list* are inserted into the same positions of *job_list* and their *machine_list* of Y, respectively. The remaining empty positions in *job_list* of Y are inserted with those jobs of X which are not in Y according to the order in which they appear in X. At the same time, all machines in *machine_list* associated with those jobs in X that participate in Y for insertion are also inserted into the same *machine_list* of their corresponding jobs in Y. It is to be noted that *nrp* is an empirical parameter.

2. *Perturbation Method*: This method itself involves two strategies. First a copy, say Y, of the solution X is created. Then both strategies are applied one by one on the solution Y. In the first strategy, two different jobs are selected randomly from *job_list*. After that these two selected jobs as well as their associated machines in their *machine_list* are swapped. This swap strategy is applied for *nswp* times, where *nswp* is the empirical parameter. Hereafter, second strategy is applied. In this strategy, a job i is selected randomly from *job_list*, then an index j of *machine_list* corresponding to the job i is selected randomly. It should be noted that an index of *machine_list* represents the operation number of the job i, i.e. O_{ij}. After that the machine k assigned for O_{ij} is replaced with an another machine k' which has the minimum processing time in its machine set of O_{ij}. It is to be noted that

if the machine set of O_{ij} has single machine, then this replacement strategy starts afresh. In addition to this one, if the selected machine k' is found to be machine k, then a different machine is randomly selected from its machine set of O_{ij} which must be different from k. This strategy is applied for at most nmc times, where nmc is an empirical parameter.

3.6 Other Features

An employed bee that associates with a food source (solution) becomes a scout bee iff this solution doesnot improve for a predetermined number of trials called *limit* parameter. When it takes place, this scout bee will generate a new solution which is similar to initialization procedure (see subsection 3.2) for generating an initial solution. As soon as a solution is generated, this scout bee again becomes employed bee. The parameter *limit* plays a vital role in ABC algorithm as it provides a balance between exploration and exploitation.

Algorithm 1 explains the pseudo-code of ABC_FJSPNW, where EP and OP denotes the number of employed and onlooker bees respectively. *Nbring_ Sol(e)* is a function that determines a solution in the neighborhood of the solution e and returns this neighboring solution. *Selection*(e_1, e_2, ..., e_{EP}) is an another function used by an onlooker bee for selecting a solution from solutions e_1, e_2, ..., e_{EP}. This function returns the index of selected solution.

4 Computational Results

ABC_FJSPNW has been implemented in C and executed on a Linux based 3.2 GHz × 4 i5 with 3.7 GB RAM system. Since to the best of our knowledge, there is no reported work for FJSPNW in the literature. So, in order to test ABC_FJSPNW we have used a set of standard benchmark instances – BRdata set [3] – which are treated as FJSPNW instances. BRdata set contains 10 instances. In this benchmark set, the number of jobs, i.e., n varies from 10 to 20, while the number of machines, i.e., m varies from 6 to 15.

Since parameters used in ABC_FJSPNW play vital roles in the performance of ABC_FJSPNW, therefore, their values are chosen carefully after a large number of experiments. Different-2 values for which parameters have been experimented are described as follows: [50, 100, 150, 200] for the total number of employed bees EP, [100, 150, 200, 250] for the total number of onlooker bees OP, [25, 50, 100] for $limit$, [0.80, 0.85, 0.90] for P_{bt}, [0.40, 0.50, 0.60] for P_r, [0.85, 0.90, 0.95] for P_{mpi}, $[0.1 \times n, 0.2 \times n, 0.3 \times n]$ for nrp, $[0.1 \times n, 0.2 \times n, 0.3 \times n]$ for $nswp$, $[0.2 \times nop, 0.3 \times nop, 0.4 \times nop]$ for nmc. nop is the total number of operations for each instance. In experimentation, the value of a single parameter is tested while keeping the values of other parameters fixed. After a large number of experimentation, it is observed that ABC_FJSPNW performs better in most of the cases when $EP = 100$, $OP = 200$, $limit = 50$, $P_{bt} = 0.85$, $P_r = 0.5$, $P_{mpi} = 0.9$, $nrp = 0.2 \times n$, $nswp = 0.2 \times n$, $nmc = 0.3 \times nop$. For each instance, ABC_FJSPNW has been executed 10 times (runs) with a different random seed. Total number of generations for each instance is $n \times m \times 30$.

Table 1. Results of ABC_FJSPNW on BRdata instances

Instance	$n \times m$	*nop*	**ABC_FJSPNW**			
			Value	Avrg.	SD	ATET
Mk01	10×6	55	45	46.90	0.83	2.26
Mk02	10×6	58	36	37.50	0.67	2.37
Mk03	15×8	150	238	250.30	6.07	22.45
Mk04	15×8	90	78	80.40	1.56	8.06
Mk05	15×4	106	247	251.20	2.93	7.83
Mk06	10×15	150	110	120.20	4.02	26.01
Mk07	20×5	100	186	189.40	2.69	10.96
Mk08	20×10	225	795	837.50	22.92	108.25
Mk09	20×10	240	620	640.00	11.63	100.67
Mk10	20×15	240	430	445.90	16.80	135.09

Experimental results of each instance obtained by ABC_FJSPNW is reported in Table 1. In Table 1, Instance denotes the name of the instance, (n, m) denotes the size of the instance, i.e., the number of jobs n and the number of machines m, *nop* presents the total number of operations associated with the instance, Value denotes the best value obtained, Avrg. denotes the average value over 10 runs, SD denotes the standard deviation, ATET represents the average total execution time in second over 10 runs. The value of SD denotes the robustness of ABC_FJSPNW for most of the instances.

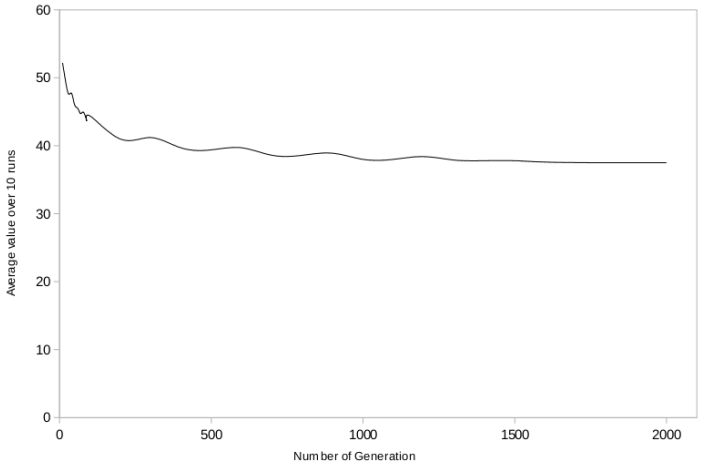


Fig. 1. Convergence behavior of ABC_FJSPNW for Mk02 instance

Fig. 1 depicts the convergence behavior of ABC_FJSPNW for the Mk02 instance. In this figure, average value over 10 runs according to the number of generations is considered. This figure shows the improvement of average solution quality of the Mk02 instance over the number of generations.

5 Conclusions

In this paper, an artificial bee colony (ABC) algorithm is proposed for a generalized problem called flexible job-shop scheduling problem with no-wait constraint (FJSPNW) combining the features of flexible job-shop scheduling problem and no-wait job-shop scheduling problem. The proposed approach has been tested on a set of standard benchmark instances for FJSPNW. Experimental results as well as convergence behavior of the proposed approach demonstrate its effectiveness.

To the best of our knowledge, there is no reported work for FJSPNW in the literature. So, as a future work, we intend to propose other metaheuristic techniques for this problem. The concepts used in this paper can also be applied to other scheduling problems.

Acknowledgements. This research is partially supported by Remanufacturing Scheduling Systems Program of A*Star in Singapore under Grant 112 290 4021.

References

1. Garey, M.R., Johnson, D.S., Sethi, R.: The Complexity of Flow Shop and Job-shop Scheduling. Mathematics of Operations Research 1, 117–129 (1976)
2. Brucker, P., Schlie, R.: Job-shop Scheduling with Multi-purpose Machines. Computing 45, 369–375 (1990)
3. Brandimarte, P.: Routing and Scheduling in a Flexible Job Shop by Tabu Search. Annals of Operations Research 41, 157–183 (1993)
4. Mastrolilli, M., Gambardella, L.M.: Effective Neighborhood Functions for the Flexible Job Shop Problem. Journal of Scheduling 3, 3–20 (2000)
5. Gao, J., Sun, L., Gen, M.: A Hybrid Genetic and Variable Neighborhood Descent Algorithm for Flexible Job Shop Scheduling Problems. Computers & Operations Research 35, 2892–2907 (2008)
6. Lenstra, J.K., Rinnooy Kan, A.H.G., Brucker, P.: Complexity of Machine Scheduling Problems. Annals of Discrete Mathematics 1, 343–362 (1977)
7. Sahni, S., Cho, Y.: Complexity of Scheduling Shops with No Wait in Process. Mathematics of Operations Research 4, 448–457 (1979)
8. Kamoun, H., Sriskandarajah, C.: The Complexity of Scheduling Jobs in Repetitive Manufacturing Systems. European Journal of Operational Research 70, 350–364 (1993)
9. Hall, N.J., Sriskandarajah, C.: A Survey on Machine Scheduling Problems with Blocking and No-wait in Process. Operations Research 44, 510–525 (1996)
10. Zhu, J., Li, X.: An Effective Meta-heuristic for No-wait Job Shops to Minimize Makespan. IEEE Transactions on Automation Science and Engineering 9, 189–198 (2012)

11. Karaboga, D.: An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report - TR06, Computer Engineering Department, Erciyes University, Turkey (2005)
12. Singh, A.: An Artificial Bee Colony Algorithm for the Leaf-Constrained Minimum Spanning Tree Problem. Applied Soft Computing 9, 625–631 (2009)
13. Pan, Q.-K., Tasgetiren, M.F., Suganthan, P.N., Chua, T.J.: A Discrete Artificial Bee Colony Algorithm for the Lot-Streaming Flow Shop Scheduling Problem. Information Sciences 181, 2455–2468 (2011)
14. Sundar, S., Singh, A.: A Swarm Intelligence Approach to the Quadratic Minimum Spanning Tree Problem. Information Sciences 180, 3182–3191 (2010)
15. Sundar, S., Singh, A.: A Swarm Intelligence Approach to the Quadratic Multiple Knapsack Problem. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) ICONIP 2010, Part I. LNCS, vol. 6443, pp. 626–633. Springer, Heidelberg (2010)
16. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A Comprehensive Survey: Artificial Bee Colony (ABC) Algorithm and Applications. Artificial Intelligence Review (2012), doi:10.1007/s10462-012-9328-0
17. Sundar, S., Singh, A.: A Swarm Intelligence Approach to the Early/Tardy Scheduling Problem. Swarm and Evolutionary Computation 4, 25–32 (2012)
18. Brizuela, C.A., Zhao, Y., Sannomiya, N.: No-wait and Blocking Job-shops: Challenging Problems for GA's. In: Proceeding of IEEE International Conference on Systems, Man, and Cybernetics, pp. 2349–2354 (2001)

Uniform Approach to Concept Interpretation, Active Contour Methods and Case-Based Reasoning

Piotr S. Szczepaniak and Arkadiusz Tomczyk

Institute of Information Technology, Lodz University of Technology
ul.Wolczanska 215, 90-924 Lodz, Poland
{piotr.szczepaniak,arkadiusz.tomczyk}@p.lodz.pl
http://www.it.p.lodz.pl

Abstract. Active contours are methods for data analysis originally developed for image segmentation. They can be treated as contextual classifiers that use expert knowledge and operate in supervised or unsupervised mode. Recently there have been developed many generalizations and extensions of those methods. One of them, proposed by the authors of this paper, reveals that they can be interpreted as methods capable of identification of more complicated structures (concepts) basing on simpler ones. In the present paper, a simple model for concept identification is presented and elucidated both in terms of active contour methods and case-based reasoning approach. The application area is any kind of data (e.g. medical images or image sequences, or even the web source data [4]) assuming they fulfill weak formal requirements.

Keywords: active contours, case-based reasoning, expert knowledge, concept interpretation.

1 Introduction

When dealing with signal-related tasks, one disposes of two types of information sources: signal data (1D, 2D or 3D) and non-signal information. The non-signal information contains knowledge about the context and can involve such information as signal acquisition parameters, the number of classes of objects expected, admissible relations between objects, etc. Depending on the task, the signal-related information can be either sufficient or insufficient for solving the given problem.

The basic tasks (usually image related) in that area are pattern recognition, classification, cluster analysis and content interpretation [1–3]. A large number of methods, including active contours [5–8], can be applied to perform those tasks. Active contours are emphasized here as their generalizations are able to deal with granules of data and contextual information. Another important example of general way for successful dealing with a number of problems of data analysis and content interpretation is the case-based reasoning (CBR) [11–13]. It uses

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 603–611, 2013.

experience collected in previously solved cases which is the main idea standing behind this method.

In the paper, the simple model of concept identification is presented. Since the concept can be interpreted as a case the integration of case-based reasoning and active contours generalizations becomes natural. The paper is organized as follows. First, the model of concept recognition is presented. In section 3, the basic active contour description using the formalism of section 2 is given. Next, case-based reasoning using the concept definition is briefly described and the necessity of expert knowledge is explained. Concluding remarks close the paper.

2 Basic Model of Concept Identification

To describe the concept identification process further the following notation, based on [14], will be used:

- N - number of considered types of concepts
- $\mathbb{L}(N) = \{1, \ldots, N\}$ - set of numerical labels assigned to the considered types of concepts
- $\mathbb{B} = \{\text{true}, \text{false}\}$ - logical values
- $\mathcal{S}$ - space of all considered concepts that can be identified, these can be initial or final concepts as well as intermediate concepts allowing to infer more complex ones
- T - type function, which is able to identify type of a given concept i.e. given a concept from $\mathcal{S}$ it assigns a proper lable from $\mathbb{L}(N)$
- $\mathcal{S}_k$ - subset of $\mathcal{S}$ with all concepts of given type $k \in \mathbb{L}(N)$
- R_k - concept recognizer, which is able to decide if a given subset of concepts from $\mathcal{S}$ represents a new concept of type $k \in \mathbb{L}(N)$ by assigning it a value from set $\mathbb{B}$
- D - decision function, determines which subsets of concepts will be used for a new concept identification
- C_k - concept function, given a proper set of concepts recognized by R_k creates a new concept of type $k \in \mathbb{L}(N)$ which is an element of set $\mathcal{S}_k$

Note that $\mathcal{S}_k = \{s \in \mathcal{S} : T(s) = k\}$. Decision function D determines which sets of concepts will be used for a new concept identification. It means that one need not to consider all the possible subsets of currently recognized concepts. On the one hand it allows to reduce the computational complexity, but on the other hand it also dermines the search space and consequently should be chosen very carefully to not exclude some crucial concepts from inference process. In practice it can directly define the considered subsets of $\mathcal{S}$ or can select its part and allow to consider all the subsets of that part. The concept recognizer R_k can be implemented in many different ways. It can simply decide which subsets of concepts can be accepted or it can evaluate their appropriateness which with some threshold can lead to similar decisions. The actual method of its work can also be varied. It can consider features of the concepts, the relations between those concepts or both those types of information which is discussed in [14]. Those two

functions compose the crucial part of the concept identification process which, within the considered model, can be described in the following way:

1. Define set $\mathcal{I} \subset \mathcal{S}$ that represents a set of initial concepts, define set $\mathcal{F} \subset \mathcal{S}$ that contains a set of final concepts, assign $i = 0$ and $\mathcal{S}(0) = \mathcal{I}$.
2. If $\mathcal{S}(i) \cap \mathcal{F} \neq \emptyset$ stop the algorithm since one of the final concepts was found.
3. By means of decision function D and concept recognizers R_k try to select a non-empty set of concepts $\mathcal{P} \in D(\mathcal{S}(i))$ such that $R_k(\mathcal{P}) = \text{true}$ for certain value of $k \in \mathbb{L}(N)$.
4. If no set satisfying the given above criteria could be found stop the algorithm.
5. Otherwise, infer a new concept $C_k(\mathcal{P})$, add it to the set of known concepts $\mathcal{S}(i+1) = \mathcal{S}(i) \cup C_k(\mathcal{P})$, assign $i = i + 1$ and repeat all the previous steps starting from the second step.

The sample interpretation of the above model and algorithm as a simplified human reasoning process and its connection with active contour methods were presented in details in [14]. In the next sections of this paper that connection will be further developed and extended by ideas coming from case-based reasoning approach used as a solution to many pattern recognition problems.

3 Active Contour Approach

The main idea of classic active contour methods is to find an optimal contour, and consequently a proper subset of pixels as it is presented in Fig. 1, in the space of all the considered contours representing certain regions in the image. The search is performed in an evolution (optimization, usually minimization) process. The quality of contour during the search is evaluated by computation of the value of certain objective function called contour energy [5–8]. It has been shown in [15] that active contours are contextual classifiers of pixels because a part of pixels (outer ones) can be interpreted as background while the rest of pixels (inner ones) belong to the object.

Energy function E usually consists of two components, external and internal, which express features of the ground and requirements about the shape of contour, respectively:

$$E = w_{ext} E_{ext} + w_{int} E_{int}$$

where w_{ext} and w_{int} are weighting coefficients. The external energy is used to evaluate localization of the contour in the image. If the contour reflects well the sought object in the image the energy is low, if not it should be high. The internal energy usually evaluates only contour without taking into account the analyzed image. Here similarly the lower value of this energy component the better contour reflects the user expectations. It can, for example, evaluate the area of the region circumscribed by the contour, the perimeter of the contour or its roundness. In general shape coefficients ought to:

a) be able to differentiate between various object shapes
b) be constant regardless of image processing transformations, such as rotation, shift and scaling
c) possess the ability of quick computation, if they are to be used in the real-time system

The definition of proper internal energy function is crucial for correct segmentation which will be presented in section 5. In practice, however, it is really hard to prepare a proper energy component. That is the place where the idea presented in section 6 becomes useful.

As it was mentioned contours are contextual classifiers of pixels (with part of the pixels belonging to the interior and the other part to the exterior of a given contour). Additionally it was shown that active contours are methods of optimal construction of classifiers [9, 15]. That interpretation allowed to generalize those methods and make them to analyze data sets composed of elements more complicated than pixels, e.g. segments of lines which can compose a high-level geometrical objects. An example of that approach was depicted in Fig. 1 where instead of pixels line segments are considered. The task is to find those segments that are dark gray and compose a circular shape. The sample energy function should of course consider the color of the segments and their configuration. For example, the possible implementation for a circle is: consideration of coordinates of the endpoints of segments of lines and definition of the concept of circle as minimum of the mean difference between the expected radius and the distance from the endpoints of segments to the expected center of the circle. Additionally, the difference between the expected perimeter and the sum of segment lengths can also be minimized. It reflects some heuristic knowledge about the concept of circle and shows that active contour approach allows one to use external domain expert knowledge or user expectation. That generalized approach was called active partitions [16–19]. The active partition name reflects the fact that similarly to active contours, that look for optimal subset of pixels, the active partitions search for optimal subsets of elements of higher granularity.

Both methods, i.e. active contours and active partitions, can be described using the formalism introduced in section 2. The following relations with one iteration of the algorithm presented there can be formulated:

- Function D defines the subsets of $\mathcal{S}$ composed of concepts of the same type (pixels, segments of lines, etc.).
- Function R_k can be implemented as an energy function with proper threshold allowing to decide whether the found contour or partition are acceptable and is of course dedicated to localize the concept of arbitrarily chosen type $k \in \mathbb{L}(N)$.
- The search for a new concept is an evolution process trying to find the optimum of energy function.
- The selected set $\mathcal{P}$ corresponds to the optimum contour or partition describing subsets of selected concepts (pixels, segments of lines, etc.).

If more than one iteration of the algorithm under consideration are taken into account one can talk about hierarchical process of image understanding where

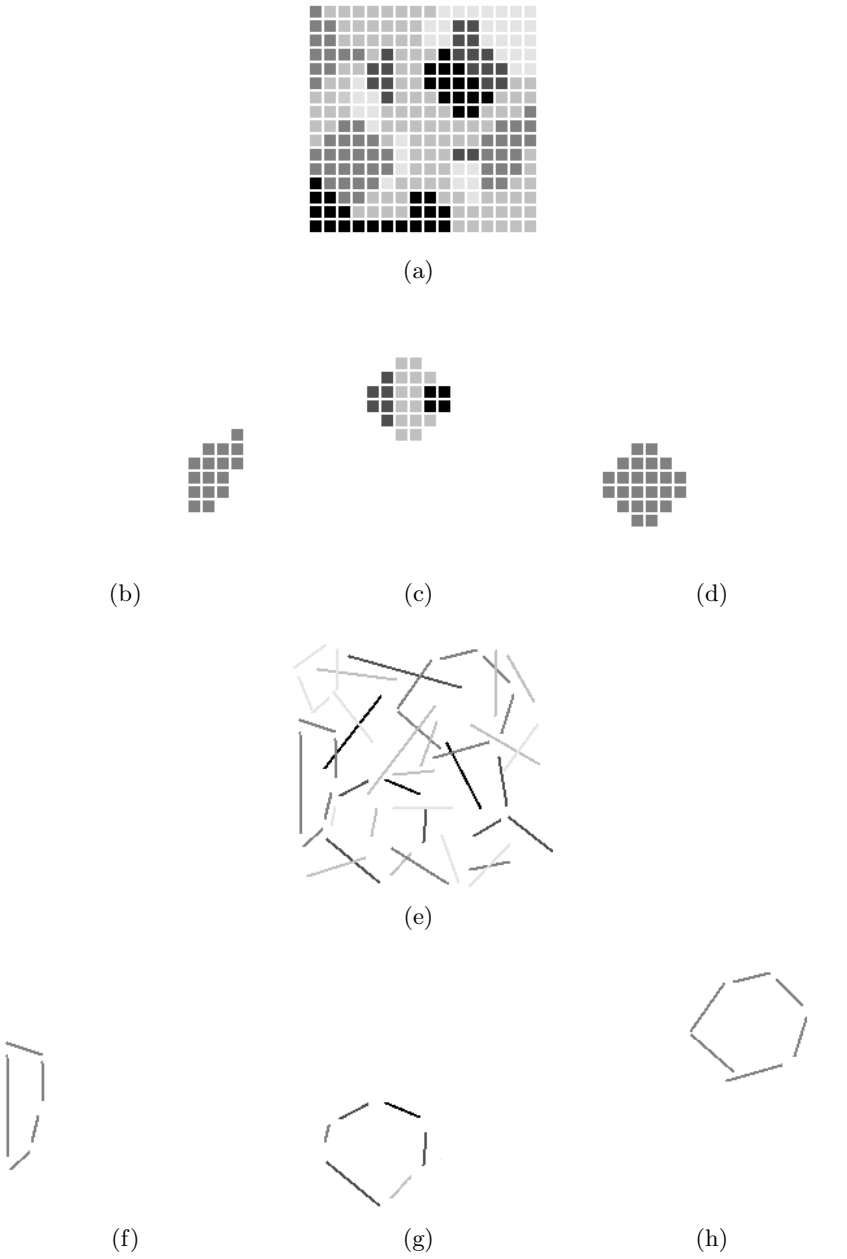


Fig. 1. An example of active contour (pixels) and active partition (segments) approach in a task of dark gray circle localization: (a) – the whole set of pixels, (b) – contour selects only subset of pixels of the same color (E_{ext}), (c) – contour selects only subset of pixels representing a desired shape (E_{int}), (d) – contour represents a proper subset of pixels (E), (e) – the whole set of segments, (f) – partition with segments of the same color (E_{ext}), (g) – partition with segments representing a desired shape (E_{int}), (h) – partition with proper segments (E)

new concepts are inferred iteratively. Those new concepts compose a knowledge that can be used during next iterations. That approach was called Cognitive Hierarchical Active Partitions (CHAP) and was examined in practice which was described in [16–19].

4 Case-Based Reasoning Principle

Case-based reasoning is considered as a method for problem solving [11–13]. When a new problem p is given, one performs the search in the memory (case base) of previously solved cases. One selects the most similar case, say p_j, and takes the associated solution s_j. This solution is either accepted in the given form or it must be modified for the new problem (Fig. 2). The search for the nearest neighbor comes from the hypothesis that similar problems have similar solutions. The importance of the concept of similarity and its proper application is evident.

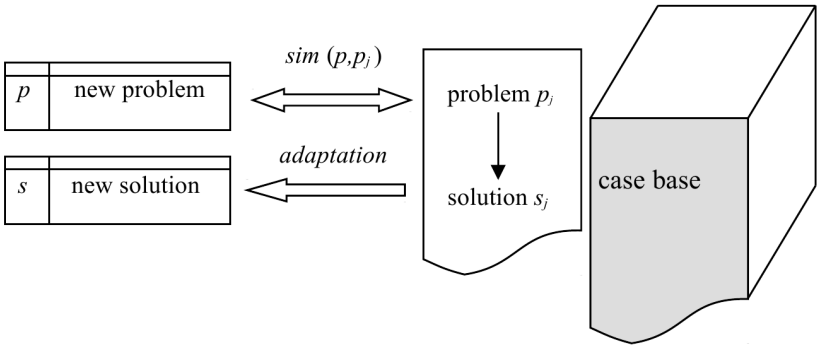


Fig. 2. The case-based reasoning principle

Within the interpretation of CBR in terms of the of model of concept identification one can argue that the case p_j and the associated solution s_j play the role of concepts, decision function D prepares the case base, the similarity measure can be considered as an element of concept recognizer R_k, and finally the process of adding a new case to the case base resembles the process of adding new concept inferred by the concept function C_k. That what differs case-based reasoning approach from the already presented concept identification is fact that method described in section 2 do not remember solutions of previous problems and do not utilize them while solving other problems of the same time. That observation can be used to extend the proposed methodology which is introduced in section 6.

5 Consideration of Expert Knowledge

As mentioned in section 3, the active contour approach allows one to use external domain expert knowledge. However, the usage of that knowledge, which can

be encoded both in energy function and in additional constraints during contour evolution, is not only an interesting possibility extending the method functionality. In practice the information in the images is often insufficient for proper understanding of the image content. For example a proper segmentation of interventricular septum requires additional knowledge about its shape. Considering only pixels with similar color leads to unsatisfactory results (Fig. 3) since whole heart muscle and tissue in the neighborhood cannot be distinguished using only that information. The proper result could be found only if in internal energy component E_{int} the information about expected shape was added.

Although the above considerations may seem to be obvious their practical application is not easy. It is especially evident in case of medical images where the diversity of object shapes causes that identification of the average shape is not be possible. A solution for that problem can be the usage of CBR principle described in section 6. Naturally all those considerations are also true for active partition approach.

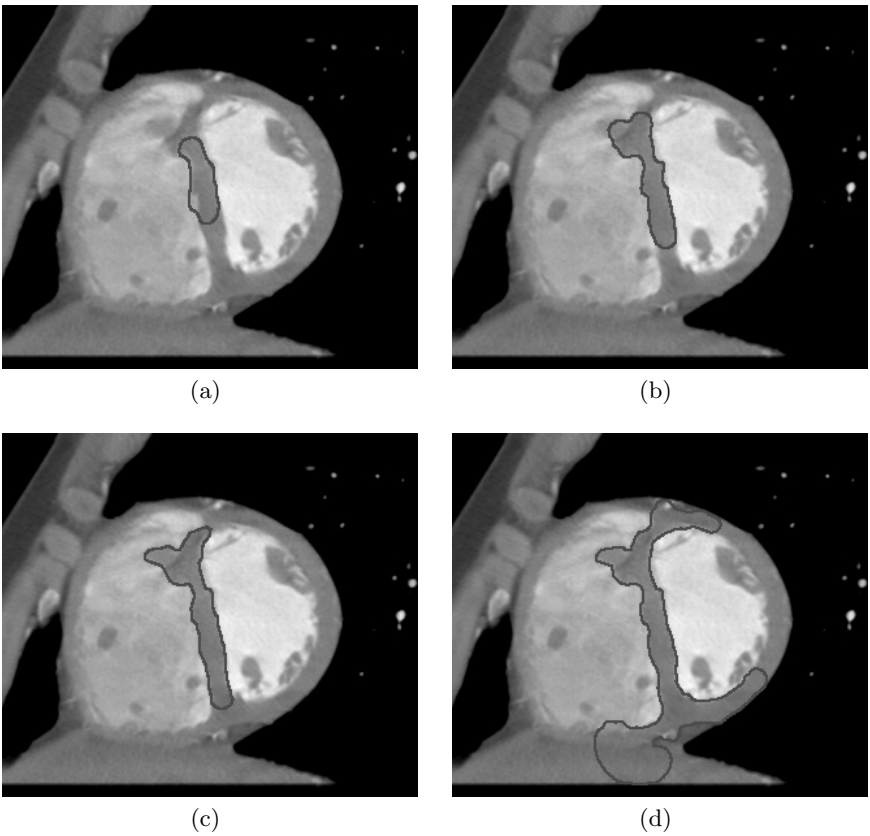


Fig. 3. Segmentation of interventricular septum with insufficient use of expert knowledge: (a), (b), (c), (d) – the selected phases of contour evolution

The additional explanation requires also the existence of additional knowledge within abstract model presented in section 2. It can be placed there in two ways. Firstly, the set of initial concepts $\mathcal{I}$ can contain additional information. Secondly, it can be used while looking for new concepts by means of decision function D and concept recognizer R_k.

6 Concluding Remarks

Due to the uniform approach to the considered methods, integration of CBR and active contours may obtain a practical form of implementation. Let us consider the problem of recognition of objects (shapes) which occur on the image to be analyzed. In CBR, the recognition of shape on the new image constitutes a new problem p. In the case base a number of images is collected for which the recognition has been correctly performed problems p_j. The features of the collected images, represented as vectors v_j, are assigned to those images, as well as the descriptions of shapes correctly recognized on the images - solutions s_j. In the elements of the vector v_j diverse information may be represented, for example statistical characteristics. For each shape s_j, a relevant active contour method can be implemented capable of searching for the similar shapes. The recognition of the shape on the new image can be performed in two stages. During the first stage, the elements of the vector v for that image are calculated. On the basis of v, the most similar case (image) p_j is found in the case base and consequently, the associated candidate solution s_j is proposed. In the second stage, the image is analyzed with the active contour method which uses in its internal energy function E_{int} the candidate shape s_j. If the value of the energy function obtained is small enough, the solution is confirmed. If not, the next closest shape is checked in terms of relevance. Again the generalization of that approach to active partitions is straightforward.

Taking into account considerations presented in previous sections those two stages within the proposed model of concept identification can be considered either as two separate iterations of the presented algorithm or as one iteration with two phases. The CBR principle gives, however, the additional possibility of the model extension. So far when a new problem was considered the previous solutions were not considered as there was no way to use previous experience. Connecting CBR with the presented model introduces the additional memory which can be present as an element of initial concepts set $\mathcal{I}$. The significance of the uniform approach is evident because it allows the conceptual and practical integration of methods having diverse roots.

References

1. Bishop, M.P.: Pattern Recognition and Machine Learning. Springer (2006)
2. Devijver, P.A., Kittler, J.: Pattern Recognition. In: A Statistical Approach, Prentice-Hall, Inc., Englewood Cliffs (1982)

3. Pedrycz, W.: Knowledge-Based Clustering. In: From Data to Information Granules. Wiley, Hoboken (2005)
4. Szczepaniak, P.S., Tomczyk, A., Pryczek, M.: Supervised Web Document Classification Using Discrete Transforms, Active Hypercontours and Expert Knowledge. In: Zhong, N., Liu, J., Yao, Y., Wu, J., Lu, S., Li, K. (eds.) Web Intelligence Meets Brain Informatics. LNCS (LNAI), vol. 4845, pp. 305–323. Springer, Heidelberg (2007)
5. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic Active Contours. Int. Journal of Computer Vision 22(1), 61–79 (2000)
6. Grzeszczuk, R., Levin, D.: Brownian Strings: Segmenting Images with Stochastically Deformable Models. IEEE Trans. on Pattern Analysis and Machine Intelligence 19(10), 100–1013 (1997)
7. Kass, M., Witkin, W., Terzopoulos, S.: Snakes: Active Contour Models. Int. Journal of Computer Vision 1(4), 321–333 (1988)
8. Tomczyk, A., Szczepaniak, P.S.: Adaptive potential active contours. Pattern Analysis and Application 14, 425–440 (2011)
9. Tomczyk, A.: Active Hypercontours and Contextual Classification. In: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications, ISDA 2005, pp. 256–261. IEEE Computer Society Press, Wroclaw (2005)
10. Tomczyk, A., Szczepaniak, P.S.: Adaptive Potential Active Hypercontours. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 692–701. Springer, Heidelberg (2006)
11. Lenz, M., Bartsch-Spörl, B., Burkhard, H.-D., Wess, S. (eds.): Case-Based Reasoning Technology. LNCS (LNAI), vol. 1400. Springer, Heidelberg (1998)
12. Pal, S.K., Dillon, T.S., Yeung, D.S. (eds.): Soft Computing in Case-Based Reasoning. Springer, Heidelberg (2001)
13. Perner, P. (ed.): Case-Based Reasoning on Images and Signals. Studies in Computational Intelligence. Springer, Heidelberg (2008)
14. Tomczyk, A., Szczepaniak, P.S.: Contribution of Active Contour Approach to Image Understanding Proceedings of the 2007 IEEE International Workshop on Imaging Systems and Techniques (IST), CD version only, ISBN: 1-3244-0965-9 (2007)
15. Tomczyk, A., Szczepaniak, P.S.: On the Relationship between Active Contours and Contextual Classification. In: Kurzyski, M., et al. (eds.) Computer Recognition Systems Proceedings of the 4th Int. Conference on Computer Recognition Systems, CORES 2005, pp. 303–310. Springer, Heidelberg (2005)
16. Tomczyk, A., Pryczek, M., Walczak, S., Jojczyk, K., Szczepaniak, P.S.: Spatch Based Active Partitions with Linguistically Formulated Energy. Journal of Applied Computer Science 18(1), 87–115 (2010)
17. Pryczek, M., Tomczyk, A., Szczepaniak, P.S.: Active Partition Based Medical Image Understanding with Self Organized, Competitive Spatch Eduction. Journal of Applied Computer Science 18(2), 67–78 (2010)
18. Jojczyk, K., Pryczek, M., Tomczyk, A., Szczepaniak, P.S., Grzelak, P.: Cognitive Hierarchical Active Partitions using Patch Approach. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2010, Part I. LNCS, vol. 6374, pp. 35–42. Springer, Heidelberg (2010)
19. Tomczyk, A., Szczepaniak, P.S., Pryczek, M.: Cognitive hierarchical active partitions in distributed analysis of medical images. Journal of Ambient Intelligence and Humanized Computing, 1–11 (2012)

Modified Merge Sort Algorithm for Large Scale Data Sets

Marcin Woźniak[1], Zbigniew Marszałek[1],
Marcin Gabryel[2], and Robert K. Nowicki[2]

[1] Institute of Mathematics, Silesian University of Technology,
ul. Kaszubska 23, 44-100 Gliwice, Poland
[2] Institute of Computational Intelligence, Czestochowa University of Technology,
Al. Armii Krajowej 36, 42-200 Czestochowa, Poland
{Marcin.Wozniak,Zbigniew.Marszalek}@polsl.pl,
{Marcin.Gabryel,Robert.Nowicki}@iisi.pcz.pl

Abstract. Sorting algorithms find their application in many fields. One of their main uses is to organize databases. Classical applications of sorting algorithms often can not cope satisfactorily with large data sets or with unfavorable poses of sorted strings. Typically, in such situations, we try to use other methods or apply sorting process to reshuffled input data. Unfortunately, this approach complicates sorting process and often results in significant prolongation of the time. In this paper, the authors examined an algorithm dedicated to the problem of sorting large scale data sets. In the literature, there are no studies of such examples. These studies will allow to describe the properties of sorting methods for large scale data sets. Performed tests have shown superior performance of the examined algorithm, especially for large scale data sets. Changes sped up sorting of data with any arrangement of the input elements.

Keywords: computer algorithm, data mining, data sorting, analysis of computer algorithms.

1 Introduction

In the literature [4, 11, 12, 24] are shown classic versions of the merge sort algorithm which are using recursion operation. The classic solution is not optimal and many researchers have proposed modifications and extensions of the basic sorting algorithms. The authors of [3, 6, 9, 10] presented the possibility of multi-threading sort algorithms. However, the authors of [2, 7, 13] presented a comparison of properties of different algorithms. The works [14, 15, 18, 21] described the impact of the algorithm on memory resources. In the paper [23] authors showed the ability to create hybrid algorithms and their implementations. The authors of the papers [8, 25–27] described the possibilities of a special matching merge algorithm. The results of the optimization algorithm for large data sets are described by the authors of [6, 21]. Recursive solutions have some limitations. The use of recursion increases time-consuming operations due to

L. Rutkowski et al. (Eds.): ICAISC 2013, Part II, LNAI 7895, pp. 612–622, 2013.

the need to operate on the stack. Processing of large collections is related to the high number of levels of recursion, which can lead to an internal stack overflow. The authors of the present study examined the merge sort algorithm for large data sets. Presented method performs the sorting process without using recursion, which increases stability and reduces the execution time. In the examined method recursive procedure for the construction of the stack was replaced by smart stacking elements into a sequence.

1.1 Classic Merge Sort Algorithm

First versions of the merge sort algorithm were published by the authors of [1, 12]. Classical algorithm known also from the literature [4, 24] splits the input string into two substrings, sorts them recursively and then merges. It uses an additional array in which are put further substrings of numbers. The procedure merges two sequences of numbers $a_p \leq \ldots \leq a_q$ and $a_{q+1} \leq \ldots \leq a_r$ stored in the array a from index p to q and from $q+1$ to r. Merging of substrings starts at the element with index $\lfloor \frac{p+r}{2} \rfloor$. Unfortunately, recursive methods, while operating on the system stack, use significantly system resources and prevent proper assessment of actual computational complexity. Moreover, in practice, for very large input strings, system stack overflow occurs frequently which leads to the suspension of the program. The solution is to use hybrid methods described in the literature, for example in [23]. However more effective solution is to use the directly acting method, which is the topic of this paper.

2 Examined Merge Sort Algorithm

The authors examined non recursive merge sort algorithm dedicated to large collections of data. Performed tests showed that changes have improved the algorithm stability measured as a reduction in dispersion (standard or average deviation) of variability of the number of clock cycles during sorting of any initial configuration of the input data. Recursive procedure for the construction of the stack, known from the literature [11, 12], was replaced by stacking items in the substrings queue. The examined algorithm has been designed to increase the efficiency of sorting large scale data sets as the algorithm based on two component algorithms. The first one is merging sorted items into the stack. The second component is a merge sorting procedure. In the first step we compare subsequent elements of the strings. As a result of the first step, we receive stacks with a given number of elements. In the second step we merge received stacks. The algorithm performs merging until there is only one present stack. Sorting algorithm will perform the procedure a sufficient number of times until there is a completely sorted input string.

2.1 Double Merge Sort Algorithm

The examined algorithm in the first step begins with a comparison of the elements in pairs of the input string $a_0, \ldots, a_{n-1}$. In this way we obtain a sequence

of pairs of elements. In the second step we merge sequences received in the first step. As a result of this operation we always receive strings containing double number of the elements. We merge these until there is only one string. If the initial string contains an odd number of elements it is sufficient to rewrite the last element till the last step in the algorithm, as shown in Fig.1. In the last step we merge this item with the string and receive completely sorted input data. Merging method receives as the input two numerical sequences $x_0 \leq \ldots \leq x_{m-1}$

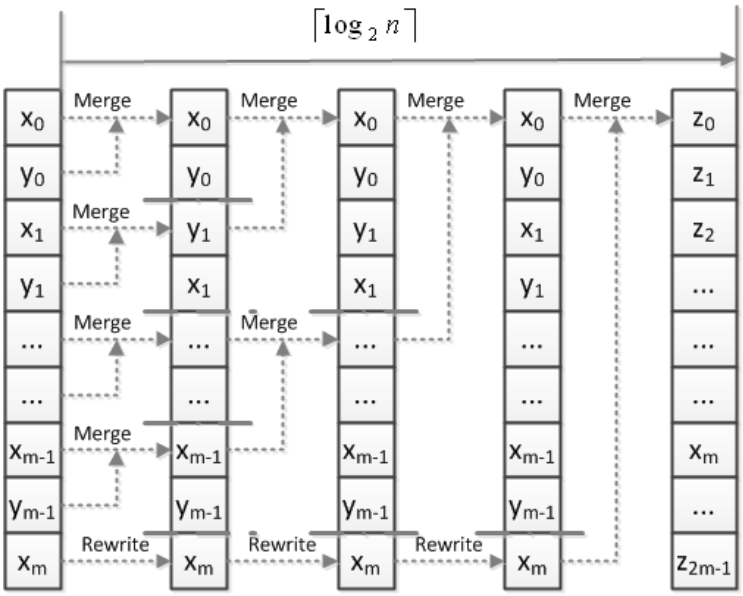


Fig. 1. An example of merging with the examined algorithm

and $y_0 \leq \ldots \leq y_{m-1}$ sorted in the previous step. It returns sorted sequence of numbers $z_0 \leq \ldots \leq z_{2m-1}$. In the following steps of the algorithm, the next element of the output string is given by selecting the smallest element of merged strings X and Y. If it happen to have the equal elements in both sequences X and Y we assume that the next selected element is the one of the string X. Due to the fact that X and Y are already sorted, we can perform merging with not more than $2m-1$ comparisons, where $2m$ is the sum of lengths of strings X and Y. The examined algorithm is based on two components presented in Algorithm 1 and Algorithm 2.

2.2 Double Merge Sort Algorithm Time Complexity

Presented algorithm of double merge sorting has time complexity described on the base of Theorem 1.

```
Begin
Load numerical sequences
Count the number of elements and write it in variables c1 i c2
1 if Strings contain elements then
    if Element of the first string is smaller than the first element of second
    string then
        Write element of the first string into the sorted string
        Cross out this item from the first string
        Decrease the number of elements in the first string
        Go to 1
    else
        Write element of the second string into the sorted string
        Cross out this item from the second string
        Decrease the number of elements in the second string
        Go to 1
    end
else
    if The first string contains elements then
        Write elements of the first string into the sorted string
        Stop
    else
        Write elements of the second string into the sorted string
        Stop
    end
end
```

Algorithm 1: Algorithm to merge sequences

Theorem 1. *Double merge sorting algorithm has time complexity*

$$T_{avg}(n) = O(n \cdot \log_2 n). \tag{1}$$

Proof. Suppose we want to sort string composed of n elements. Performing examined double merge sort algorithm we compare in each step sequences arranged in the previous step. In the first step we merge 1-element sequences which we compare in successive pairs. Thus, in this iteration, we make no more than n comparisons. In the next iteration we merge strings from the first step. As a result of the merge operation in step two we have in order strings of doubled elements. Such an arrangement is obtained as a result of no more than n comparisons. Merging continues in subsequent iterations respectively comparing each time strings of a double length, resulted from consolidation in the previous step. If we compare two sequences of length m we will do no more than $2m - 1$ comparisons. Every time by merging such two strings we obtain a double sequence. To sort the input string we will do k merge steps in the algorithm, and in each of them we do no more than n comparisons. Without loss of generality, we can

```
   Start
   Load sorted string into a table a
   Count number of elements and write it in the variable n
   Set logical variable t of direction on true
   Create space for array b with size of the number of sorted elements n
   Set size of merged series as 1 and write it in a variable m
1  if Size of merged series is smaller than n then
       Set index of first merged string as 0 and write it in variable i
2      if i is smaller than n then
           Set index pb of merged string on i
           Set index p1 of first merged string on i
           Set index p2 of second merged string on i + m
           if index of second string p2 is smaller or equal to n then
               Set number of elements in first string c1 as m
               Set number of elements in second string c2 as n − p2
               if Number of elements in second string is bigger than m then
                   Set number of elements of second string c2 as m
                   Go to 3
               else
                   Go to 3
               end
3              if Check if logical variable t is set on true then
                   Proceed Algorithm to merge sequences from array a and
                   write them into array b
                   Increase index i of first string by 2 ∗ m
                   Go to 2
               else
                   Proceed Algorithm to merge sequences from array b and
                   write them into array a
                   Increase index i of the first string by 2 ∗ m
                   Go to 2
               end
           else
               Increase index i of the first string by 2 ∗ m
               Go to 2
           end
       else
           Change logical variable t of direction to opposite
           Double size of merged series
           Go to 1
       end
   else
       if Negation of direction variable is set on true then
           Copy elements from array b to array a
           Stop
       else
           Stop
       end
   end
```

Algorithm 2: Algorithm of merge sorting

assume that the sorted sequence of n elements has approximately 2^k compared elements. Thus, it is reasonable to estimate the following

$$\min_{k \in N} 2^k \geq n. \tag{2}$$

Logarithms both sides (2) we obtain

$$\min_{k \in N} \log_2 2^k \geq \log_2 n. \tag{3}$$

Thus, on the basis of a logarithmic function we obtain the following inequality

$$\min_{k \in N} k \cdot \log_2 2 \geq \log_2 n. \tag{4}$$

Finally, we can assume that the number of operations performed by sorting the string will be

$$k = \lceil \log_2 n \rceil . \tag{5}$$

So time complexity is

$$T_{avg}(n) = n \cdot k = n \cdot \log_2 n, \tag{6}$$

which completes our discussion. □

2.3 Results of Egzaminations

The examined algorithm for sorting by double merging has been tested to evaluate it's performance for large scale data sets. The method was programmed using standard CLR in MS Visual Studio 2010 Ultimate. To tests were taken random samples of 100 series in each class of frequencies, including unfavorable positioning. Tests were carried out on processor i7 series. During performed tests were analyzed CPU clock cycles. Statistical analysis of the results is presented in Table 1. The speed of performed operations depends on the initial arrangement of the input data. It is also affected by the number of changes made at a constant number of comparisons. In this study, the authors have examined the values shown in Table 1. Experiments were performed for a sorted string, inversely sorted string and other numerical sequences taken at random. Table 1 shows statistical study of the CPU clock cycles during the tests of the algorithm described in Section 2.1. Results presented in Table 1 were plotted. On the charts are also plotted results of similar experiments carried out for the base algorithm. The aim of the analysis and comparison is to verify the thesis that the changes improved large scale data sets sorting. In the examinations and tests were compared the characteristics of the examined algorithm with the version known from the literature [4, 11, 12, 24]. The analysis of Figure 2 shows that the examined method behaves similarly to the classic version of the algorithm. Size of the average number of CPU clock cycles shown in Figure 2 is related to the standard deviation shown in Figure 3. Charts shown in Figure 3 show the comparison of the characteristics of the standard deviation of CPU clock cycles.

Table 1. Table of CPU clock cycles of the examined double merge sorting algorithm

cpu tics [ti]	Number of sorted elements			
	100	1000	10000	100000
avg	2298, 4	2987, 6	8631, 2	32754, 4
std deviation	801, 397	265, 39	2619, 49	10721, 64
avg deviation	472, 16	215, 52	2139, 44	9266, 08
coef. of variation	0, 35	0, 09	0, 30	0, 33
var. area upper end	1497	2722, 21	6011, 71	22032, 76
var. area lower end	3099, 8	3252, 99	11250, 69	43476, 04
	1000000	10000000	100000000	
avg	363782, 8	4353284	48195771, 8	
std deviation	125670, 76	1384543, 28	15067068, 72	
avg deviation	110078, 16	1211305, 2	13203157, 76	
coef. of variation	0, 34	0, 32	0, 31	
var. area upper end	238112, 04	2968740, 72	33128703, 08	
var. area lower end	489453, 56	5737827, 28	63262840, 52	

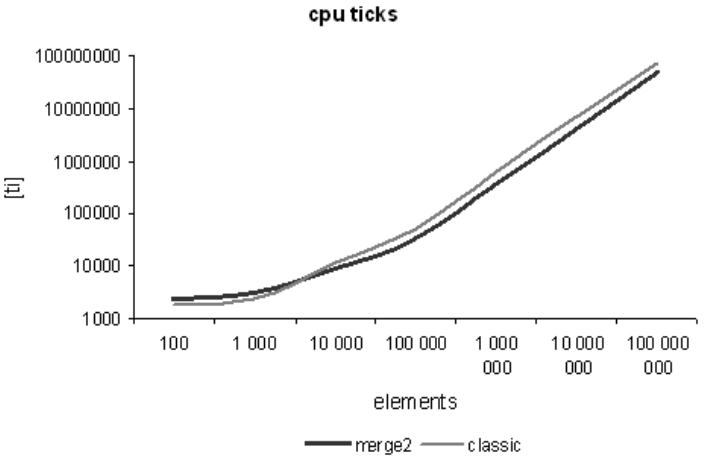


Fig. 2. Comparing the values of the characteristics of CPU clock cycles

Graphs shown in Figure 3 show that the examined method has a positive influence on sorting large scale data sets. At the same time the algorithm described in Section 2.1 is characterized by increased stability. Coefficients of variation were approximied by polynomial approximation, which is shown in Figure 4. The resulting curves characterize variability of characteristics of merge sort algorithm. An analysis of the chart shown in Figure 5 shows that classical form is efficient only for series smaller than 1000 elements. Above this number of items, described in Section 2.1 algorithm runs about 40% faster than the classic version. At the same time a comparison of the values shown in Figure 3 shows that it is working with more stability than the classic version. Figure 5 shows that the examined

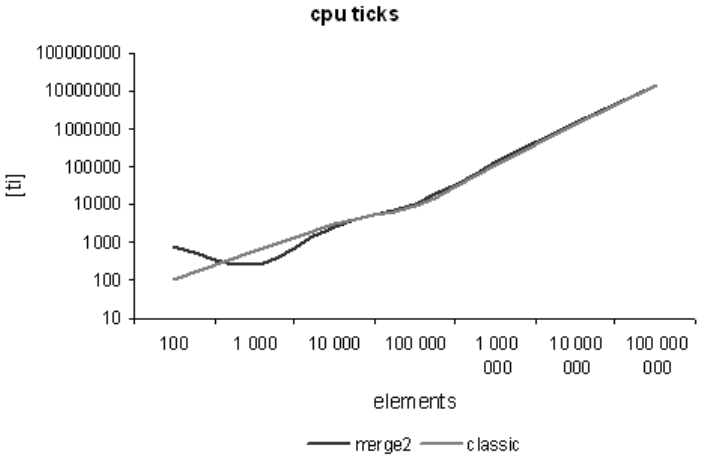


Fig. 3. Comparing the values of standard deviation of CPU clock cycles

Table 2. Comparative table of CPU clock cycles variations of the classic and the examined algorithm

cpu tics [ti]	Number of sorted elements						
	100	1000	10000	100000	1000000	10000000	100000000
classic	0, 05	0, 25	0, 24	0, 18	0, 36	0, 2	0, 19
merge2	0, 35	0, 09	0, 3	0, 33	0, 34	0, 32	0, 31

double merge sort can effectively sort any set of elements. Simultaneously, as shown by the earlier analysis, it has greater stability. The analysis and comparison of charts in Figure 5 shows that the examined method can sort efficiently any set of elements, which confirms previous conclusions. Described in Section 2.1 algorithm behaves more stable and sort quickly any collection of large data regardless of the orientation of sorted items.

3 Final Remarks

In conclusion, presented double merge sort algorithm has a good stability for large scale data sets. It is also a fast version for sorting large input data of any arrangement of the elements. For described in Section 2.1 algorithm, during the experiments there were no difficulties in sorting adverse poses of elements in the input series. The presented method allows sorting any data sets and increases the stability in comparison to the classical form. This effect is mainly due to the lack of handling the return stack. Thus, the examined algorithm appears to be appropriate for use for large scale data sets. However authors consider whether it is possible to increase the efficiency of sorting large data sets. Further research and work will focus on improving the speed and stability. In further

work on increasing the efficiency the authors plan to focus on building and improving multiple merging algorithm. An important change may be to abandon the classical method 'divide and conquer' for increased distribution of the stack.

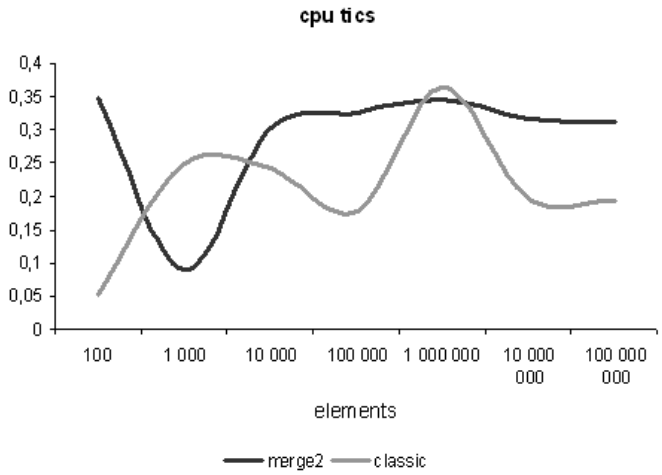


Fig. 4. Comparing the values of expected variability of CPU clock cycles

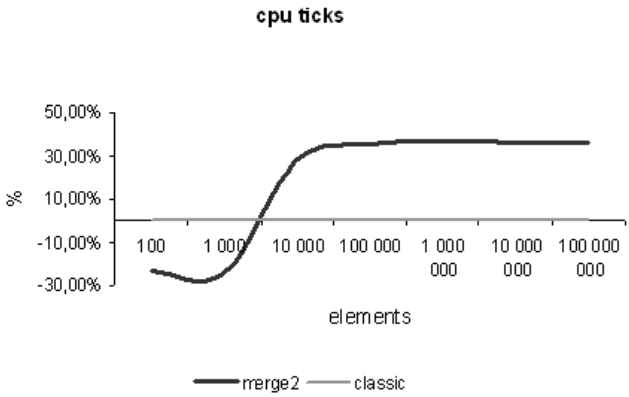


Fig. 5. Illustration of the percentage difference of CPU clock cycles between classic version and examined algorithm

Described in Section 2.1 double merge sort algorithm is a stable method, especially for collections of over 10000 elements, as shown in Figure 4. The question is, what is the difference between the classic version and described in Section 2.1 algorithm. Comparison of the characteristics is illustrated in Figure 5. As the reference method is chosen classic algorithm.

References

1. Aho, I.A., Hopcroft, J., Ullman, J.: The design and analysis of computer algorithms. Addison-Wesley Publishing Company, USA (1975)
2. Blelloch, G.E., Leiserson, C.E., Maggs, B.M., Plaxton, C.G., Smith, S.J., Zagha, M.: A comparison of sorting algorithms for the connection machine CM-2. In: Proceedings of the 3rd Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA 1991), Hilton Head, South Carolina, pp. 3–16 (July 1991)
3. Cole, R.: Parallel merge sort. SIAM J. Comput. 17, 770–785 (1988)
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to algorithms. The MIT Press and McGraw-Hill Book Company, Cambridge (2001)
5. Crescenzi, P., Grossi, R., Italiano, G.F.: Search data structures for skewed strings. Experimental and Efficient Algorithms, Second International Workshop, WEA 2003, Ascona, Switzerland. In: Jansen, K., Margraf, M., Mastrolli, M., Rolim, J.D.P. (eds.) WEA 2003. LNCS, vol. 2647, pp. 81–96. Springer, Heidelberg (2003)
6. Dlekmann, R., Gehring, J., Luling, R., Monien, B., Nubel, M., Wanka, R.: Sorting large data sets on a massively parallel system. In: Proceedings of the 6th Symposium on Parallel and Distributed Processing, pp. 2–9. IEEE, Los Alamitos (1994)
7. Estivill-Castro, V., Wood, D.: A Survey of Adaptive Sorting Algorithms. Computing Surveys 4(24), 441–475 (1992)
8. Gedigaa, G., Duntschb, I.: Approximation quality for sorting rules. Computational Statistics & Data Analysis 40, 499–526 (2002)
9. Helman, D.R., Bader, D.A., JaJa, J.: A Randomized Parallel Sorting Algorithm with an Experimental Study. Parllel and Dirtributed Computing 1(52), l-23 (1998)
10. Jeon, M.S., Kim, D.S.: Parallel Merge Sort with Load Balancing. International Journal of Parallel Programming 1(31), 21–33 (2003)
11. Kruse, R.L., Ryba, A.J.: Data Structures and Program Design in C++, 2nd edn. Pearson Education (1999)
12. Knuth, D.E.: Sorting and Searching, 2nd edn. The Art of Computer Programming, vol. 3. Addison-Wesley, Reading (1998)
13. Larriba-Pey, J.: An Analysis of Superscalar Sorting Algorithms on an R8000 Processor. In: Intl. Conf. of the Chilean Computing Society, Chile, pp. 125–134 (1997)
14. LaMarca, A., Ladner, R.E.: The influence of caches on the performance of sorting. In: Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, Louisiana, January 5-7, pp. 370–379 (1997)
15. LaMarca, A., Ladner, R.E.: The Influence of Caches on the Performance of Sorting. In: Proc. Eighth Ann. ACM-SIAM Symp. Discrete Algorithms (1997)
16. Larson, P.: External Sorting: Run Formation Revisited. IEEE Transactionson Knowledge and Data Engineering 15(4), 961–972 (2003)
17. Shi, H., Schaeffer, J.: Parallel sorting by regular sampling. Journal of Parallel and Distributed Computing 4(14), 361–372
18. Pai, V.S., Varman, P.J.: Prefetching with Multiple Disks for External Mergesort: Simulation and Analysis. In: Proc. Int. Conf. Data Eng., pp. 273–282 (1992)
19. Raghaven, P.: Lecture Notes on Randomized Algorithms, tech. report, IBM Research Division, Yorktown Heights, New York (1990)
20. Rashid, L., Hassanein, W.M., Hammad, M.A.: Analyzing and Enhancing the Parallel Sort Operation on Multithreaded Architectures. J. Supercomputer (2009)
21. Salzberg, B.: Merging Sorted Runs Using Large Main Memory. Acta Informatica 27(3), 195–215 (1989)

22. Sinha, R., Zobel, J.: Cache-conscious sorting of large sets of strings with dynamic tries. J. Exp. Algorithmics 9, 1–5 (2004)
23. Trimananda, R., Haryanto, C.Y.: A Parallel Implementation of Hybridized Merge-Quicksort Algorithm on MPICH. In: 2010 International Conference on Distributed Framework for Multimedia Applications (DFmA)
24. Weiss, M.A.: Data Structure & Algorithm Analysis in C++, 2nd edn. Addison-Wesley Longman (1999)
25. Zhang, W., Larson, P.A.: Dynamic Memory Adjustment for External Mergesort. In: Proc. Very Large Data Bases Conf., pp. 376–385 (1997)
26. Zhang, W., Larson, P.A.: Buffering and Read-Ahead Strategies for External Mergesort. In: Proc. Very Large Data Bases Conf., pp. 523–533 (1998)
27. Zheng, L., Larson, P.A.: Speeding Up External Mergesort. IEEE Trans. Knowledge and Data Eng. 8(2), 322–332 (1996)

Author Index

Adedoyin-Olowe, Mariam II-135
Adrjanowicz, Lukasz I-515, II-275
Alexandre, Enrique I-503
Álvarez, Lorena I-503, I-568
Amrhein, Wolfgang II-122
Askari, Mohsen I-215

Balicki, Jerzy II-477
Bartczuk, Łukasz I-227
Bazan, Marek I-20
Białka, Szymon II-167
Bilski, Jarosław I-32
Blachnik, Marcin II-167
Bobulski, Janusz I-515
Bożejko, Wojciech II-489
Burczyński, Tadeusz I-345, II-79, II-569
Burghardt, Andrzej I-64, II-351
Buza, Krisztian I-437

Camalich, Juan II-407
Castermans, Thierry II-251
Castiello, Ciro I-318
Chua, T.J. II-593
Cierniak, Robert I-524
Coelho, Clarimar José II-24
Cpałka, Krzysztof I-329, I-493, II-91, II-113, II-342, II-523
Cuadra, Lucas I-503
Czajkowski, Marcin II-1
Czarnowski, Ireneusz I-41

Daniel, Milan I-235
da Silva, Alberione Braz I-183
da Silva Soares, Anderson II-24
Davendra, Donald II-56
de Lima, Telma Woerle II-24
Dembczyński, Krzysztof II-500
Długosz, Adam II-79
Dobosz, Piotr I-580
Drozda, Paweł I-532
Duda, Piotr I-459
Dudek, Grzegorz I-52
Dudek-Dyduch, Ewa II-419
Dutkiewicz, Lidia II-419
Duvinage, Matthieu II-251

Dymova, Ludmila I-247
Dziatkiewicz, Grzegorz II-569
Dziwiński, Piotr I-227

Edwards, Kieran Jay II-146
Ernst, Sebastian II-363

Fanelli, Anna Maria I-318
Fevens, Thomas II-263

Gaber, Mohamed Medhat II-135, II-146
Gabryel, Marcin I-540, II-12, II-612
Gadamer, Marcin I-76
Gałkowski, Tomasz II-158
Gaweł, Przemysław II-500
Gil-Pita, Roberto I-503, I-568
Godziuk, Andrzej I-559
Górecki, Przemysław I-532
Grzegorzewski, Przemysław II-441
Guijarro-Berdiñas, Bertha I-162
Guitouni, Adel II-535

Habibzadeh, Mehdi II-263
Hajduk, Zbigniew I-282
Hayashi, Yoichi I-1
Hendzel, Zenon I-64, II-351
Hetmaniok, Edyta II-431
Horzyk, Adrian I-76

Isshiki, Masaharu I-108

Jankowski, Michal I-548
Jaszkiewicz, Andrzej II-500
Jaworski, Bartosz II-101
Jaworski, Maciej I-459
Jędrzejowicz, Piotr I-41
Jelonkiewicz, Jerzy I-88
Jiménez, A. I-400

Kacprzak, Dariusz I-259
Kacprzak, Magdalena I-271
Kacprzyk, Janusz I-150, I-483
Kaczor, Krzysztof II-453, II-511
Kapruziak, Mariusz I-559
Keith, Tūreiti I-96
Kempa, Wojciech M. II-12

Kinoshita, Koji I-108
Klement, Erich Peter II-122
Klęsk, Przemysław I-559
Klimek, Radosław II-363
Kluska, Jacek I-118, I-282
Kluza, Krzysztof II-453
Knaś, Michał I-580
Knop, Michal I-524
Kolendo, Piotr II-101
Kominkova Oplatkova, Zuzana II-56
Koppelstätter, Werner II-122
Kordos, Mirosław II-167
Korytkowski, Marcin I-540, I-588
Kosiński, W. Konrad I-259
Kosiński, Witold I-259, I-271, I-388
Kotłowski, Wojciech II-500
Kotulski, Leszek II-397
Kowalski, Piotr A. I-294
Krawczak, Maciej II-176
Krawczyk, Bartosz I-426
Krawczyk, Henryk II-477
Kretowski, Marek II-1
Krichen, Saoussen II-535
Krzyżak, Adam II-263
Kubanek, Mariusz II-275
Kuczkowski, Lukasz II-101
Kukier, Marta I-483
Kulczycki, Piotr II-184
Kuś, Wacław II-79, II-569
Kusy, Maciej I-118

Ładyżyński, Piotr II-441
Łapa, Krystian I-329, II-113, II-523
Laskowski, Łukasz I-88
Laureano, Gustavo Teodoro II-24
Ławryńczuk, Maciej I-130
Lee, Kin-Hong II-330
Lee, Kwan-Yeung II-330
Leśniak, Paweł II-320
Leung, Kwong-Sak II-330
Li, Jianchan I-215
Litwiński, Sławomir II-309
Llerena, Cosme I-503, I-568
Lorent, Anna I-580
Lower, Michał I-306
Lucarelli, Marco I-318
Lughofer, Edwin II-122
Łukasik, Szymon II-184

Magott, Jan I-306
Mańdziuk, Jacek II-547
Marszałek, Adam I-345
Marszałek, Zbigniew II-612
Marussy, Kristóf I-437
Masri, Hela II-535
Mateos, A. I-400
Mazurkiewicz, Jacek I-548
McLean, David II-375
Mealing, Richard II-385
Mencar, Corrado I-318
Michalak, Marcin II-194
Mleczko, Wojciech K. I-448
Mrugalski, Marcin I-140
Murakami, Kenji I-108

Najgebauer, Patryk I-588, I-602, I-613, II-309
Nalepa, Grzegorz J. II-453, II-511
Nowak, Bartosz A. I-448
Nowak, Tomasz I-588, I-602, II-309
Nowicki, Robert K. I-448, II-12, II-612

Okulewicz, Michał II-547
Olech, Bogdan I-559
Olszewski, Dominik I-150
Orzechowski, Patryk II-206
Oszust, Mariusz II-218

Pabiasz, Sebastian II-286
Pasupa, Kitsuchart II-297
Patan, Krzysztof II-320
Peteiro-Barral, Diego I-162
Piech, Henryk II-559
Piersa, Jarosław I-205
Pietraszek, Jacek I-353
Pietruczuk, Lena I-459
Pleszczyński, Mariusz II-431
Płoński, Piotr I-474
Pluciński, Marcin II-230
Pluhacek, Michal II-36, II-56
Poteralski, Arkadiusz II-79, II-569
Prokopowicz, Piotr I-365
Przybył, Andrzej I-227, II-91, II-523
Ptak, Aleksandra II-559

Rafajłowicz, Ewaryst I-193
Rafajłowicz, Wojciech II-46
Riche, Nicolas II-251

Romanowski, Jakub I-588, I-613, II-309
Rosa, João Luís Garcia I-183
Rosa-Zurera, Manuel I-568
Rusiecki, Andrzej I-174
Rutkowska, Danuta I-412
Rutkowski, Grzegorz II-320
Rutkowski, Leszek I-540
Rybakov, Vladimir II-375
Rygał, Janusz I-588, I-602, I-613
Rymko, Łukasz II-477

Salvini, Rogério Lopes II-24
Samali, Bijan I-215
Sawai, Hidefumi II-465
Scherer, Rafał I-540, I-602, I-613
Sędziwy, Adam II-397
Senkerik, Roman II-36, II-56
Sevastjanov, Pavel I-247
Shapiro, Jonathan L. II-385
Skorupski, Jacek I-306
Skubalska-Rafajłowicz, Ewa I-20, I-193
Skulimowski, Andrzej M.J. II-582
Słoń, Grzegorz I-376
Słoński, Marek I-624
Słota, Damian II-431
Slowik, Adam II-67
Śmierzchalski, Roman II-101
Smoląg, Jacek I-32
Smorawa, Dorota II-275
Sopyła, Krzysztof I-532
Soto, Alexandro II-407
Soto, Ramon II-407
Stahl, Frederic II-135
Starczewski, Artur II-242
Starczewski, Janusz T. II-286
Starosta, Bartłomiej I-388
Stawarz, Magdalena II-194
Suganthan, P.N. II-593
Sundar, Shyam II-593
Szarecki, Adam II-500
Szczepaniak, Piotr S. II-603
Szczepanik, Mirosław II-79, II-569
Szczypta, Jacek II-91
Sze-To, Ho-Yin II-330
Szkatuła, Grażyna II-176
Szmidt, Eulalia I-483
Szuster, Marcin I-64, II-351
Szymański, Julian II-477

Tang, Nelson L.S. II-330
Tekieli, Marcin I-624
Tkacz, Kamil I-247
Tomczyk, Arkadiusz II-603
Tso, Kai-Yuen II-330

Uchroński, Mariusz II-489

Vicente, E. I-400

Weddell, Stephen J. I-96
Węgrzyn-Wolska, Katarzyna I-271
Weidenholzer, Günther II-122
Wiaderek, Krzysztof I-412
Witkowska, Anna II-101
Wodecki, Mieczysław II-489
Wojnicki, Igor II-363
Wong, Man-Hon II-330
Woźniak, Marcin II-12, II-612
Woźniak, Michał I-426
Wysocki, Marian II-218

Zadrożny, Sławomir I-150
Zając, Katarzyna I-205
Zalasiński, Marcin I-329, I-493, II-113, II-342
Zaremba, Krzysztof I-474
Zăvoianu, Alexandru-Ciprian II-122
Żbikowski, Kamil II-441
Zelinka, Ivan II-36, II-56
Zielonka, Adam II-431