

Piotr Jędrzejowicz
Ngoc Thanh Nguyen
Kiem Hoang (Eds.)

LNAI 6922

Computational Collective Intelligence

Technologies and Applications

Third International Conference, ICCCI 2011
Gdynia, Poland, September 2011
Proceedings, Part I

1
Part I

 Springer

Lecture Notes in Artificial Intelligence 6922

Subseries of Lecture Notes in Computer Science

LNAI Series Editors

Randy Goebel

University of Alberta, Edmonton, Canada

Yuzuru Tanaka

Hokkaido University, Sapporo, Japan

Wolfgang Wahlster

DFKI and Saarland University, Saarbrücken, Germany

LNAI Founding Series Editor

Joerg Siekmann

DFKI and Saarland University, Saarbrücken, Germany

Piotr Jędrzejowicz Ngoc Thanh Nguyen
Kiem Hoang (Eds.)

Computational Collective Intelligence

Technologies and Applications

Third International Conference, ICCCI 2011
Gdynia, Poland, September 21-23, 2011
Proceedings, Part I

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

Piotr Jędrzejowicz
Gdynia Maritime University
Morska 81-87
81-225 Gdynia, Poland
E-mail: pj@am.gdynia.pl

Ngoc Thanh Nguyen
Wrocław University of Technology
Wyb. Wyspińskiego 27
50-370 Wrocław, Poland
E-mail: ngoc-thanh.nguyen@pwr.wroc.pl

Kiem Hoang
University of Information Technology
Km 20, Xa Lo Ha Noi, Linh Trung, Thu Duc
848 HCM City, Vietnam
E-mail: kiemhv@uit.edu.vn

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-23934-2 e-ISBN 978-3-642-23935-9
DOI 10.1007/978-3-642-23935-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011935853

CR Subject Classification (1998): I.2, I.2.11, H.3-4, C.2, D, H.5, K.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

© Springer-Verlag Berlin Heidelberg 2011

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

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

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

Printed on acid-free paper

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

Preface

Computational Collective Intelligence – Technologies and Applications Third International Conference ICCCI 2011 September 21–23, 2011, Gdynia, Poland

This volume contains the proceedings (Part I) of the Third International Conference on Computational Collective Intelligence (ICCCI 2011) held at Gdynia Maritime University in Poland during September 21–23, 2011. The conference was organized by Gdynia Maritime University in cooperation with Wrocław University of Technology in Poland. The conference was run under the scientific patronage of the Committee of Informatics, Polish Academy of Sciences and the Polish Artificial Intelligence Society.

Following the successes of the First International Conference on Computational Collective Intelligence: Semantic Web, Social Networks and Multiagent Systems (ICCCI 2009) held in Wrocław, Poland, and the Second International Conference on Computational Collective Intelligence (ICCCI 2010) held in Kaohsiung, Taiwan, ICCCI 2011 continued to provide an internationally respected forum for scientific research in the computer-based methods of collective intelligence and their applications.

Computational collective intelligence (CCI) is most often understood as a sub-field of artificial intelligence (AI) dealing with soft computing methods that enable group decisions to be made or knowledge to be processed among autonomous units acting in distributed environments. Methodological, theoretical and practical aspects of CCI are considered as the form of intelligence that emerges from the collaboration and competition of many individuals (artificial and/or natural). The application of multiple computational intelligence technologies such as fuzzy systems, evolutionary computation, neural systems, consensus theory, etc., can support human and other collective intelligence, and create new forms of CCI in natural and/or artificial systems. Three subfields of application of computational intelligence technologies to support various forms of collective intelligence are of special attention but are not exclusive: Semantic Web (as an advanced tool increasing collective intelligence), social network analysis (as the field targeted to the emergence of new forms of CCI), and multiagent systems (as a computational and modeling paradigm especially tailored to capture the nature of CCI emergence in populations of autonomous individuals).

The ICCCI 2011 conference featured a number of keynote talks, oral presentations and invited sessions, closely aligned to the theme of the conference. The conference attracted a substantial number of researchers and practitioners from

all over the world, who submitted their papers for the main track subdivided into ten thematic streams and seven special sessions.

The main track streams, covering the methodology and applications of CCI, include: Machine Learning and Applications, Collective Computations and Optimization, Web Services and Semantic Web, Social Networks, Complex Systems and Intelligent Applications, Ontology Management, Knowledge Management, Agents and Multiagent Systems, Mobile Agents and Robotics, Modeling, Simulation and Decision Making, Applications of Computational Collective Intelligence in Shipping. The special sessions, covering some specific topics of particular interest, include: Computational Collective Intelligence in Bioinformatics, Computational Collective Intelligence-Based Optimization Models, Autonomous and Collective Decision-Making, Collective Intelligence in Web Systems, Web Systems Analysis, Computational Swarm Intelligence and Applications, Computational Swarm Intelligence, Discovering Relationships in Data, and finally, Computational Collective Intelligence in Economy.

We received almost 300 submissions from over 25 countries. Each paper was reviewed by two to four members of the International Program Committee and International Reviewer Board. Only 109 best papers were selected for oral presentation and publication in the two volumes of the ICCCI 2011 proceedings.

We would like to express our sincere thanks to the Honorary Patrons: the Mayor of Gdynia, Wojciech Szczurek, the Rector of Gdynia Maritime University, Romuald Cwilewicz, and the Rector of Wrocław University of Technology, Tadeusz Więckowski. Our special gratitude goes to the Honorary Chairs, Pierre Lévy from the University of Ottawa, Canada, and Roman Słowiński from Poznań University of Technology, Poland, for their support.

We would also like to express our thanks to the Keynote Speakers: Jeng-Shyang Pan, Leszek Rutkowski, Edward Szczerbicki and Jan Treur, for their interesting and informative talks of world-class standard. We also thank our partners, University of Information Technology (Vietnam), National Taichung University of Education (Taiwan), and Academic Computer Centre in Gdansk (Poland), for their kind support.

Special thanks go to the Organizing Chairs (Radosław Katarzyniak and Dariusz Barbucha) for their efforts in the organizational work. Thanks are due to the Program Co-chairs, Program Committee and the Board of Reviewers, essential for reviewing the submissions to ensure the high quality of accepted papers. We also thank the members of the Local Organizing Committee, Publicity Chairs and Special Sessions Chairs.

Finally, we cordially thank all the authors, presenters and delegates for their valuable contribution to this successful event. The conference would not have been possible without their support.

It is our pleasure to announce that the ICCCI conference series is closely cooperating with the Springer journal *Transactions on Computational Collective Intelligence* and the IEEE SMC Technical Committee on *Transactions on Computational Collective Intelligence*.

We hope that ICCCI 2011 significantly contributed to the fulfillment of the academic excellence, leading to even more successful of ICCCI events in the future.

September, 2011

Piotr Jędrzejowicz
Ngoc Thanh Nguyen
Kiem Hoang

Organization

Honorary Patrons

Wojciech Szczurek	Mayor of Gdynia, Poland
Romuald Cwilewicz	Rector of Gdynia Maritime University
Tadeusz Więckowski	Rector of Wrocław University of Technology

Honorary Chairs

Pierre Lévy	University of Ottawa, Canada
Roman Słowiński	Poznan University of Technology, Poland

General Chairs

Piotr Jędrzejowicz	Gdynia Maritime University, Poland
Ngoc Thanh Nguyen	Wrocław University of Technology, Poland

ICCCI Steering Committee

Ngoc Thanh Nguyen	Wrocław University of Technology, Poland – Chair
Piotr Jędrzejowicz	Gdynia Maritime University, Poland – Co-chair
Ryszard Kowalczyk	Swinburne University of Technology, Australia – Co-chair
Shyi-Ming Chen	National Taiwan University of Science and Technology, Taiwan
Adam Grzech	Wrocław University of Technology, Poland
Lakhmi C. Jain	University of South Australia, Australia
Geun-Sik Jo	Inha University, Korea
Janusz Kacprzyk	Polish Academy of Sciences, Poland
Ryszard Tadeusiewicz	AGH University of Science and Technology, Poland
Toyooki Nishida	Kyoto University, Japan

Program Chairs

Ireneusz Czarnowski	Gdynia Maritime University, Poland
Jason J. Jung	Yeungnam University, Korea
Ryszard Kowalczyk	Swinburne University of Technology, Australia
Kazumi Nakamatsu	University of Hyogo, Japan

Organizing Chairs

Dariusz Barbucha Gdynia Maritime University, Poland
Radosław Katarzyniak Wrocław University of Technology, Poland

Special Session Chairs

Amine Chohra Paris-East University, France
Tokuro Matsuo Yamagata University, Japan
Ewa Ratajczak-Ropel Gdynia Maritime University, Poland

Publicity Chair

Izabela Wierzbowska Gdynia Maritime University, Poland

Doctoral Track Chair

Bogdan Trawiński Wrocław University of Technology, Poland

Keynote Speakers

Jeng-Shyang Pan
National Kaohsiung University of Applied Sciences, Taiwan
Overview of Algorithms for Swarm Intelligence

Leszek Rutkowski
Technical University of Częstochowa, Poland
Rough-Neuro-Fuzzy-Genetic Hybrid Intelligent Systems

Edward Szczerbicki
The University of Newcastle, Australia
Experiential Decisional DNA

Jan Treur
VU University Amsterdam, The Netherlands
From Mirroring to the Emergence of Shared Understanding and Collective Power

Special Sessions

1. *Computational Collective Intelligence in Bioinformatics (CCIB 2011)*

Stanisław Kozielski	Silesian University of Technology, Poland
Bożena Małyśiak-Mrozek	Silesian University of Technology, Poland
Dariusz Mrozek	Silesian University of Technology, Poland

2. *CCI-Based Optimization Models (CCIBOM 2011)*

Piotr Jędrzejowicz	Gdynia Maritime University, Poland
Dariusz Barbucha	Gdynia Maritime University, Poland

3. *Autonomous and Collective Decision-Making (ACDM 2011)*

Amine Chohra	Paris-East University, France
--------------	-------------------------------

4. *Collective Intelligence in Web Systems—Web Systems Analysis (WebSys 2011)*

Kazimierz Choroś	Wrocław University of Technology, Poland
Mohamed Hassoun	ENSSIB, Villeurbanne, France

5. *Computational Collective Intelligence in Economy (CCIE 2011)*

Tadeusz Szuba	AGH University of Science and Technology, Poland
Stanisław Szydło	AGH University of Science and Technology, Poland
Paweł Skrzyński	AGH University of Science and Technology, Poland

6. *Swarm Intelligence and Applications (SIA 2011)*

Mong-Fong Horng	National Kaohsiung University of Applied Sciences, Taiwan
Jeng-Shyang Pan	National Kaohsiung University of Applied Sciences, Taiwan

7. *Computational Swarm Intelligence—Discovering Relationships in Data (CSI 2011)*

Urszula Boryczka	University of Silesia, Poland
Mariusz Boryczka	University of Silesia, Poland
Marcin Budka	Bournemouth University, UK
Katarzyna Musiał	Bournemouth University, UK

International Program Committee

Costin Badica	University of Craiova, Romania
Youcef Baghdadi	Sultan Qaboos University, Oman
Dariusz Barbucha	Gdynia Maritime University, Poland
František Čapkovič	Slovak Academy of Sciences, Slovakia
Hsuan-Ting Chang	National Yunlin University of Science and Technology, Taiwan
Rung-Ching Chen	Chaoyang University of Technology, Taiwan
Shyi-Ming Chen	National Taichung University of Education, Taiwan
Yuh-Ming Cheng	Shu-Te University, Taiwan
Amine Chochra	Paris-East University, France
Ireneusz Czarnowski	Gdynia Maritime University, Poland
Phuc Do	University of Information Technology, Vietnam
Mauro Gaspari	University of Bologna, Italy
Daniela Godoy	Unicen University, Argentina
Kiem Hoang	University of Information Technology, Vietnam
Tzung-Pei Hong	National University of Kaohsiung, Taiwan
Wen-Lian Hsu	Academia Sinica, Taiwan
Feng-Rung Hu	National Taichung University of Education, Taiwan
Jingshan Huang	University of South Alabama, USA
Dosam Hwang	Yeungnam University, Korea
Gordan Jezic	University of Zagreb, Croatia
Joanna Jędrzejowicz	University of Gdańsk, Poland
Piotr Jędrzejowicz	Gdynia Maritime University, Poland
Joanna Józefowska	Poznan University of Technology, Poland
Jason J. Jung	Yeungnam University, Korea
Janusz Kacprzyk	Polish Academy of Sciences, Poland
Andrzej Kasprzak	Wroclaw University of Technology, Poland
Radosław Katarzyniak	Wroclaw University of Technology, Poland
Muhammad Khurram Khan	King Saud University, Kingdom of Saudi Arabia
Bor-Chen Kuo	National Taichung University of Education, Taiwan
Halina Kwaśnicka	Wroclaw University of Technology, Poland
Chin-Feng Lee	Chaoyang University of Technology, Taiwan
Xiaofeng Li	Texas A&M University, USA
Hsiang-Chuan Liu	Asia University, Taiwan
Tokuro Matsuo	Yamagata University, Japan
Kazumi Nakamatsu	University of Hyogo, Japan
Ngoc Thanh Nguyen	Wroclaw University of Technology, Poland
Manuel Núñez	Universidad Complutense de Madrid, Spain
Tarkko Oksala	Helsinki University of Technology, Finland
Cezary Orłowski	Gdańsk University of Technology, Poland

Jeng-Shyang Pan	National Kaohsiung University of Applied Sciences, Taiwan
Kunal Patel	Ingenuity Systems, USA
Witold Pedrycz	University of Alberta, Canada
Ramalingam Ponnusamy	Aarupadai Veedu Institute of Technology, India
Ewa Ratajczak-Ropel	Gdynia Maritime University, Poland
Quanzheng Sheng	University of Adelaide, Australia
Tian-Wei Sheu	National Taichung University of Education, Taiwan
Janusz Sobacki	Wroclaw University of Technology, Poland
Bogdan Trawiński	Wroclaw University of Technology, Poland
Rainer Unland	University of Duisburg-Essen, Germany
Sheng-Yuan Yang	St. John's University, Taiwan
Yunming Ye	Harbin Institute of Technology, China

International Referee Board

Ouahiba Azouaoui	Radomil Matousek
Mariusz Boryczka	Alina Momot
Urszula Boryczka	Dariusz Mrozek
Leszek Borzemski	Katarzyna Musiał
Krzysztof Brzostowski	Mahamed G.H. Omran
Marcin Budka	Chung-Ming Ou
Bohdan S. Butkiewicz	Paweł Pawlewski
Krzysztof Cetnarowicz	Andrzej Polański
Yun-Heh (Jessica) Chen-Burger	Panrasee Ritthipravat
Tzu-Fu Chiu	Ewa Romuk
Amine Chohra	Przemysław Różewski
Kazimierz Choroś	Joanna Rzeszowska
Krzysztof Cyran	Andrzej Siemiński
Jarosław Drapała	Aleksander Skakovski
Jan Tadeusz Duda	Paweł Skrzyński
Trong Hai Duong	Jacek Stańdo
Włodzimierz Filipowicz	Chaoli Sun
Paulina Golińska	Joanna Szlarczyńska
Sylwia Górczyńska-Kosiorz	Tadeusz Szuba
Mong-Fong Horng	Jerzy Tiuryn
Jacek Kabziński	Chun-Wei Tseng
Jarosław Janusz Kacerka	Leuo-hong Wang
Arkadiusz Kawa	Waldemar Wieczerzycki
Muhammad Khurram Khan	Andrzej Wierniak
Stanisław Kozielski	Izabela Wierzbowska
Ondrej Krejcar	Aleksander Zgrzywa
Andrei Lihu	Quan Zou
Bożena Małysiak-Mrozek	

Table of Contents – Part I

Keynote Speeches

From Mirroring to the Emergence of Shared Understanding and Collective Power	1
<i>Jan Treur</i>	
Experiential Knowledge in the Development of Decisional DNA (DDNA) and Decisional Trust for Global e-Decisional Community	17
<i>Edward Szczerbicki and Cesar Sanin</i>	
Overview of Algorithms for Swarm Intelligence	28
<i>Shu-Chuan Chu, Hsiang-Cheh Huang, John F. Roddick, and Jeng-Shyang Pan</i>	

Machine Learning and Applications

Neural Network Committees Optimized with Evolutionary Methods for Steel Temperature Control	42
<i>Miroslaw Kordos, Marcin Blachnik, Tadeusz Wieczorek, and Slawomir Golak</i>	
Growing Hierarchical Self-Organizing Map for Images Hierarchical Clustering	52
<i>Bartłomiej M. Buczek and Paweł B. Myszkowski</i>	
AdaBoost Ensemble of DCOG Rough–Neuro–Fuzzy Systems	62
<i>Marcin Korytkowski, Robert Nowicki, Leszek Rutkowski, and Rafał Scherer</i>	
A Two-Armed Bandit Collective for Exemplar Based Mining of Frequent Itemsets with Applications to Intrusion Detection	72
<i>Vegard Haugland, Marius Kjølleberg, Svein-Erik Larsen, and Ole-Christoffer Granmo</i>	
Applications of Paraconsistent Artificial Neural Networks in EEG	82
<i>Jair Minoro Abe, Helder F.S. Lopes, Kazumi Nakamatsu, and Seiki Akama</i>	
Features Selection in Character Recognition with Random Forest Classifier	93
<i>Władysław Homenda and Wojciech Lesiński</i>	

Generating and Postprocessing of Biclusters from Discrete Value Matrices	103
<i>Marcin Michalak and Magdalena Stawarz</i>	
A Validity Criterion for Fuzzy Clustering	113
<i>Stanisław Brodowski</i>	
Estimations of the Error in Bayes Classifier with Fuzzy Observations . . .	123
<i>Robert Burduk</i>	
Building Context-Aware Group Recommendations in E-Learning Systems	132
<i>Danuta Zakrzewska</i>	
Investigation of Random Subspace and Random Forest Methods Applied to Property Valuation Data	142
<i>Tadeusz Lasota, Tomasz Luczak, and Bogdan Trawiński</i>	
Application of Data Mining Techniques to Identify Critical Voltage Control Areas in Power System	152
<i>Robert A. Lis</i>	
 Collective Computations and Optimization	
Linkage Learning Based on Local Optima	163
<i>Hamid Parvin and Behrouz Minaei-Bidgoli</i>	
Data Extrapolation and Decision Making via Method of Hurwitz-Radon Matrices	173
<i>Dariusz Jakóbczak</i>	
The Memetic Ant Colony Optimization with Directional Derivatives Simplex Algorithm for Time Delays Identification	183
<i>Janusz P. Papliński</i>	
Advanced Prediction Method in Efficient MPC Algorithm Based on Fuzzy Hammerstein Models	193
<i>Piotr M. Marusak</i>	
Evolutionary Tuning of Compound Image Analysis Systems for Effective License Plate Recognition	203
<i>Krzysztof Krawiec and Mateusz Nawrocki</i>	
Investigation of Self-adapting Genetic Algorithms Using Some Multimodal Benchmark Functions	213
<i>Magdalena Smętek and Bogdan Trawiński</i>	
Multiobjective Particle Swarm Optimization Using Fuzzy Logic	224
<i>Hossein Yazdani, Halina Kwaśnicka, and Daniel Ortiz-Arroyo</i>	

An Evolutionary Algorithm for the Urban Public Transportation	234
<i>Jolanta Koszelew</i>	
Exploring Market Behaviors with Evolutionary Mixed-Games Learning Model	244
<i>Yu Du, Yingsai Dong, Zengchang Qin, and Tao Wan</i>	
Web Services and Semantic Web	
On the Web Ontology Rule Language OWL 2 RL	254
<i>Son Thanh Cao, Linh Anh Nguyen, and Andrzej Szalas</i>	
Results of Research on Method for Intelligent Composing Thematic Maps in the Field of Web GIS	265
<i>Piotr Grobelny and Andrzej Pieczyński</i>	
OAuth+UAO: A Distributed Identification Mechanism for Triplestores	275
<i>Dominik Tomaszuk and Henryk Rybiński</i>	
Propagating and Aggregating Trust with Uncertainty Measure	285
<i>Anna Stachowiak</i>	
On Ordered Weighted Reference Point Model for Multi-attribute Procurement Auctions	294
<i>Bartosz Kozłowski and Włodzimierz Ogryczak</i>	
ASPARAGUS - A System for Automatic SPARQL Query Results Aggregation Using Semantics	304
<i>Agnieszka Lawrynowicz, Jędrzej Potoniec, Łukasz Konieczny, Michał Madziar, Aleksandra Nowak, and Krzysztof T. Pawlak</i>	
Protégé Based Environment for DL Knowledge Base Structural Analysis	314
<i>Mariusz Chmielewski and Piotr Stapor</i>	
Fuzzy Reliability Analysis of Simulated Web Systems	326
<i>Tomasz Walkowiak and Katarzyna Michalska</i>	
Using Multi-attribute Structures and Significance Term Evaluation for User Profile Adaptation	336
<i>Agnieszka Indyka-Piasecka</i>	
A Method for Web-Based User Interface Recommendation Using Collective Knowledge and Multi-attribute Structures	346
<i>Michał Malski</i>	

Social Networks

Opinion Analysis from the Social Web Contributions	356
<i>Kristína Machová</i>	
Modelling Trust for Communicating Agents: Agent-Based and Population-Based Perspectives	366
<i>S. Waqar Jaffry and Jan Treur</i>	
Multidimensional Social Network: Model and Analysis	378
<i>Przemysław Kazienko, Katarzyna Musiał, Elżbieta Kukla, Tomasz Kajdanowicz, and Piotr Bródka</i>	
Modelling and Simulation of an Infection Disease in Social Networks . . .	388
<i>Rafał Kasprzyk, Andrzej Najgebauer, and Dariusz Pierzchała</i>	
Distributed Military Simulation Augmented by Computational Collective Intelligence	399
<i>Dariusz Pierzchała, Michał Dyk, and Adam Szydlowski</i>	
Time Based Modeling of Collaboration Social Networks	409
<i>Gabriel Tutoky and Ján Paralič</i>	
Simulating Riot for Virtual Crowds with a Social Communication Model	419
<i>Wei-Ming Chao and Tsai-Yen Li</i>	

Complex Systems and Intelligent Applications

Building Detection and 3D Reconstruction from Two-View of Monocular Camera	428
<i>My-Ha Le and Kang-Hyun Jo</i>	
Design of an Energy Consumption Scheduler Based on Genetic Algorithms in the Smart Grid	438
<i>Junghoon Lee, Gyung-Leen Park, Ho-Young Kwak, and Hongbeom Jeon</i>	
Toward Cyclic Scheduling of Concurrent Multimodal Processes	448
<i>Grzegorz Bocewicz, Robert Wójcik, and Zbigniew A. Banaszak</i>	
Meteorological Phenomena Forecast Using Data Mining Prediction Methods	458
<i>František Babič, Peter Bednár, František Albert, Ján Paralič, Juraj Bartók, and Ladislav Hluchý</i>	
Artificial Immune Clustering Algorithm to Forecasting Seasonal Time Series	468
<i>Grzegorz Dudek</i>	

Knowledge-Based Pattern Recognition Method and Tool to Support Mission Planning and Simulation	478
<i>Ryszard Antkiewicz, Andrzej Najgebauer, Jarosław Rulka, Zbigniew Tarapata, and Roman Wantoch-Rekowski</i>	
Secure UHF/HF Dual-Band RFID : Strategic Framework Approaches and Application Solutions	488
<i>Namje Park</i>	
Kernel PCA in Application to Leakage Detection in Drinking Water Distribution System	497
<i>Adam Nowicki and Michał Grochowski</i>	
Decisional DNA Digital TV: Concept and Initial Experiment	507
<i>Haoxi Zhang, Cesar Sanin, and Edward Szczerbicki</i>	
Application of Program Agents for Optimisation of VoIP Communication	517
<i>Hrvoje Očevčić and Drago Žagar</i>	
Study of Diabetes Mellitus (DM) with Ophthalmic Complication Using Association Rules of Data Mining Technique	527
<i>Pornnapas Kasemthaweesab and Werasak Kurutach</i>	
Intelligent Management Message Routing in Ubiquitous Sensor Networks	537
<i>Junghoon Lee, Gyung-Leen Park, Hye-Jin Kim, Cheol Min Kim, Ho-Young Kwak, Sang Joon Lee, and Seongjun Lee</i>	
On Ranking Production Rules for Rule-Based Systems with Uncertainty	546
<i>Beata Jankowska and Magdalena Szymkowiak</i>	
Smart Work Workbench; Integrated Tool for IT Services Planning, Management, Execution and Evaluation	557
<i>Mariusz Fraś, Adam Grzech, Krzysztof Juszczyzyn, Grzegorz Kotaczek, Jan Kwiatkowski, Agnieszka Prusiewicz, Janusz Sobocki, Paweł Świątek, and Adam Wasilewski</i>	
Ontology Management	
A Cut-Free ExpTime Tableau Decision Procedure for the Description Logic SHI	572
<i>Linh Anh Nguyen</i>	
IT Business Standards as an Ontology Domain	582
<i>Adam Czarnecki and Cezary Orłowski</i>	

Attribute Selection-Based Recommendation Framework for Long-Tail User Group: An Empirical Study on MovieLens Dataset	592
<i>Jason J. Jung and Xuan Hau Pham</i>	
IOEM - Ontology Engineering Methodology for Large Systems	602
<i>Joanna Śliwa, Kamil Gleba, Wojciech Chmiel, Piotr Szwed, and Andrzej Głowacz</i>	
A Framework for Building Logical Schema and Query Decomposition in Data Warehouse Federations	612
<i>Rafał Kern, Krzysztof Ryk, and Ngoc Thanh Nguyen</i>	
A Distance Function for Ontology Concepts Using Extension of Attributes' Semantics	623
<i>Marcin Pietranik and Ngoc Thanh Nguyen</i>	
Author Index	633

Table of Contents – Part II

Knowledge Management

Some Properties of Complex Tree Integration Criteria	1
<i>Marcin Maleszka and Ngoc Thanh Nguyen</i>	
Semantically Enhanced Collaborative Filtering Based on RSVD	10
<i>Andrzej Szwabe, Michał Ciesielczyk, and Tadeusz Janasiewicz</i>	
Hybrid Recommendation Based on Low-Dimensional Augmentation of Combined Feature Profiles	20
<i>Andrzej Szwabe, Tadeusz Janasiewicz, and Michał Ciesielczyk</i>	
Statement Networks Development Environment <i>REx</i>	30
<i>Wojciech Cholewa, Tomasz Rogala, Paweł Chrzanowski, and Marcin Amarowicz</i>	
Domain Based Semantic Compression for Automatic Text Comprehension Augmentation and Recommendation	40
<i>Dariusz Ceglarek, Konstanty Haniewicz, and Wojciech Rutkowski</i>	
Model of Community-Build System for Knowledge Development	50
<i>Przemysław Różeński</i>	

Agents and Multi-agent Systems, Mobile Agents and Robotics

A Multi-Agent Scheduling Approach for the Joint Scheduling of Jobs and Maintenance Operations in the Flow Shop Sequencing Problem	60
<i>Si Larabi Khelifati and Fatima Benbouzid-Sitayeb</i>	
Aligning Simple Modalities in Multi-agent System	70
<i>Wojciech Lorkiewicz, Grzegorz Popek, Radostaw Katarzyniak, and Ryszard Kowalczyk</i>	
Multilateral Negotiations in Distributed, Multi-agent Environment	80
<i>Piotr Pałka</i>	
Route Guidance System Based on Self Adaptive Multiagent Algorithm	90
<i>Mortaza Zolfpour Arokhlo, Ali Selamat, Siti Zaiton Mohd Hashim, and Md Hafiz Selamat</i>	
Agent-Based System with Learning Capabilities for Transport Problems	100
<i>Bartłomiej Śnieżyński and Jarosław Koźlak</i>	

Modelling of Agents Cooperation and Negotiation	110
<i>František Čapkovič</i>	
Modelling Relationship between Antecedent and Consequent in Modal Conditional Statements	120
<i>Grzegorz Skorupa and Radosław Katarzyniak</i>	
Semantic Simulation Engine for Supervision of Mobile Robotic System	130
<i>Janusz Będkowski and Andrzej Mastowski</i>	
Cognitive Supervision and Control of Robotic Inspection-Intervention System	140
<i>Janusz Będkowski and Andrzej Mastowski</i>	
Declarative Design of Control Logic for Mindstorms NXT with XTT2 Method	150
<i>Grzegorz J. Nalepa and Błażej Biesiada</i>	
Modeling, Simulation and Decision Making	
Planning in Collaborative Stigmergic Workspaces	160
<i>Constantin-Bala Zamfirescu and Ciprian Candea</i>	
Signature Verification Based on a Global Classifier That Uses Universal Forgery Features	170
<i>Joanna Putz-Leszczyńska and Andrzej Pacut</i>	
Functional and Dependability Approach to Transport Services Using Modelling Language	180
<i>Katarzyna Michalska and Jacek Mazurkiewicz</i>	
Swarm-Based Multi-agent Simulation: A Case Study of Urban Traffic Flow in the City of Wrocław	191
<i>Dariusz Król and Maciej Mrozek</i>	
Evolving Equilibrium Policies for a Multiagent Reinforcement Learning Problem with State Attractors	201
<i>Florin Leon</i>	
Agent Based Simulation of Customers Behavior for the Purpose of Price Distribution Estimation	211
<i>Marek Zachara and Cezary Piskor-Ignatowicz</i>	

Applications of Computational Collective Intelligence in Shipping

Evolutionary Sets of Safe Ship Trajectories: Problem-Dedicated Operators	221
<i>Rafał Szlapczyński and Joanna Szlapczyńska</i>	
Evolutionary Sets of Safe Ship Trajectories: Improving the Method by Adjusting Evolutionary Techniques and Parameters	231
<i>Rafał Szlapczyński</i>	
Comparison of Selection Schemes in Evolutionary Method of Path Planning	241
<i>Piotr Kolendo, Bartosz Jaworski, and Roman Śmierzchalski</i>	
Evidence Representation and Reasoning in Selected Applications	251
<i>Włodzimierz Filipowicz</i>	
Application of Artificial Intelligence Methods for the Diagnosis of Marine Diesel Engines	261
<i>Adam Charchalis and Rafał Pawletko</i>	

Computational Collective Intelligence in Bioinformatics

Scalable System for Protein Structure Similarity Searching	271
<i>Bożena Małysiak-Mrozek, Alina Momot, Dariusz Mrozek, Lukasz Hera, Stanisław Kozielski, and Michał Momot</i>	
Efficient Algorithm for Microarray Probes Re-annotation	281
<i>Paweł Foszner, Aleksandra Gruca, Andrzej Polański, Michał Marczyk, Roman Jaksik, and Joanna Polańska</i>	

CCI-Based Optimization Models

Learning Method for Co-operation	290
<i>Ewa Dudek-Dyduch and Edyta Kucharska</i>	
Experimental Evaluation of the Agent-Based Population Learning Algorithm for the Cluster-Based Instance Selection	301
<i>Ireneusz Czarnowski and Piotr Jędrzejowicz</i>	
Double-Action Agents Solving the MRCPSP/Max Problem	311
<i>Piotr Jędrzejowicz and Ewa Ratajczak-Ropel</i>	
Parallel Cooperating A-Teams	322
<i>Dariusz Barbucha, Ireneusz Czarnowski, Piotr Jędrzejowicz, Ewa Ratajczak-Ropel, and Izabela Wierzbowska</i>	

Solving the Capacitated Vehicle Routing Problem by a Team of Parallel Heterogeneous Cooperating Agents	332
<i>Dariusz Barbuca</i>	

Autonomous and Collective Decision-Making

Validated Decision Trees versus Collective Decisions	342
<i>Krzysztof Grąbczewski</i>	

Time and Personality Dependent Behaviors for Agent Negotiation with Incomplete Information	352
<i>Amine Chohra, Arash Bahrammirzaee, and Kurosh Madani</i>	

Dynamic Selection of Negotiation Protocol in Multi-agent Systems for Disaster Management	363
<i>Amelia Bădică, Costin Bădică, Sorin Ilie, Alex Muscar, and Mihnea Scafes</i>	

Collective Intelligence in Web Systems - Web Systems Analysis

Guaranteeing Quality of Service in Globally Distributed Web System with Brokers	374
<i>Krzysztof Zatwarnicki</i>	

Customized Travel Information Recommendation Framework Using CBR and Collective Intelligence	385
<i>Mye Sohn, Su ho Kang, and Young Min Kwon</i>	

Integration of Collective Knowledge in Fuzzy Models Supporting Web Design Process	395
<i>Jaroslav Jankowski</i>	

WordNet Based Word Sense Disambiguation	405
<i>Andrzej Siemiński</i>	

Further Tests with Click, Block, and Heat Maps Applied to Website Evaluations	415
<i>Kazimierz Choroś</i>	

A Research Study on Business-Oriented Quality-Driven Request Service in a B2C Web Site	425
<i>Grażyna Suchacka and Leszek Borzemski</i>	

Computational Collective Intelligence in Economy

Collective Intelligence Approach to Measuring Invisible Hand of the Market	435
<i>Paweł Skrzyński, Tadeusz Szuba, and Stanisław Szydło</i>	

Collective Intelligence of Genetic Programming for Macroeconomic Forecasting	445
<i>Jerzy Duda and Stanisław Szydło</i>	

Computational Swarm Intelligence and Applications

Parallel Appearance-Adaptive Models for Real-Time Object Tracking Using Particle Swarm Optimization	455
<i>Bogusław Rymut and Bogdan Kwolek</i>	

Following the Leader – Particle Dynamics in Constricted PSO	465
<i>Jacek Kabziński</i>	

Computational Swarm Intelligence - Discovering Relationships in Data

An Adaptive Discretization in the ACDT Algorithm for Continuous Attributes	475
<i>Urszula Boryczka and Jan Kozak</i>	

Approximate Nash Equilibria in Bimatrix Games	485
<i>Urszula Boryczka and Przemysław Juszcuk</i>	

Co-operative, Parallel Simulated Annealing for the VRPTW	495
<i>Rafał Skinderowicz</i>	

The Parallel Ant Vehicle Navigation System with CUDA Technology ...	505
<i>Wojciech Bura and Mariusz Boryczka</i>	

Author Index	515
---------------------------	-----

From Mirroring to the Emergence of Shared Understanding and Collective Power

Jan Treur

VU University Amsterdam, Agent Systems Research Group
De Boelelaan 1081, 1081 HV, Amsterdam, The Netherlands
treur@cs.vu.nl
<http://www.cs.vu.nl/~treur>

Abstract. Mirror neurons and internal simulation are core concepts in the new discipline Social Neuroscience. In this paper it is discussed how such neurological concepts can be used to obtain social agent models. It is shown how these agent models can be used to obtain emergence of shared understanding and collective power of groups of agents, both in a cognitive and affective sense.

1 Introduction

In human society in many situations some form of ‘sharedness’ or ‘collectiveness’ is experienced, which often covers cognitive as well as affective dimensions. Although this is a very common type of phenomenon, at forehand it is not at all clear how it can emerge. For example, the experience of feeling good being part of a group with a shared understanding and collective action may occur as quite natural. However, as persons in a group are autonomous agents with their own neurological structures and patterns, carrying, for example, their own emotions, beliefs, and intentions, it would be more reasonable to expect that such sharedness and collectiveness is impossible to achieve. Nevertheless, often groups develop coherent views and decisions, and, even more surprisingly, the group members seem to share a good feeling with it. This process depends on possibilities for (informational, motivational and emotional) transfer between individuals, which can be enhanced, for example, by social media.

In recent years new light has been shed on this seeming paradox by developments in neuroscience, in particular, in the new discipline called Social Neuroscience; e.g., [9], [10], [19], [20], [28]. Two interrelated core concepts in this discipline are mirror neurons and internal simulation. Mirror neurons are neurons that not only have the function to prepare for a certain action or body change, but are also activated upon observing somebody else who is performing this action or body change; e.g., [34], [45], [51]. Internal simulation is internal processing that copies processes that may take place externally, for example, in another individual; e.g., [13], [15], [24], [26], [30].

In this paper, first in Section 2 some core concepts from Social Neuroscience are briefly reviewed. Next, in Section 3 it is discussed how based on them shared understanding can emerge. This covers both cognitive and affective understanding, and in a combined form empathic understanding. In Section 4 it is discussed how

collective decisions and actions may emerge, and how such collective actions can be grounded in a shared affective understanding. Section 5 illustrates how such phenomena can be formalised in computational models. Finally, Section 6 is a discussion.

2 Mirror Neurons, Internal Simulation and Social Neuroscience

In this section two core concepts in the discipline of Social Neuroscience are briefly discussed: mirror neurons and internal simulation. Together they realise an individual's mental function of mirroring mental processes of another individual.

2.1 The Discovery of Mirror Neurons

Recently it has been found that in humans a specific type of neurons exists, called *mirror neurons*, which both are active to prepare for certain actions or bodily changes and when such actions or body states are observed in other persons. The discovery of mirror neurons originates from single cell recording experiments with monkeys in Parma in the 1990s. In particular, the focus was on an area in the premotor cortex (F5) known to be involved in the preparation of grasp actions. By accident, and to their own surprise, the researchers discovered that some of the recorded cells were not only firing when the monkey was preparing a grasp action, but also when somebody in the lab was grasping something and the monkey just observed that; cf. [23], [48]; see also [34], [50], [51]. The highly unexpected element was that recognition of observed actions of others involves the subject's preparation for the same type of action. It turned out that in the premotor area F5 about 20% of the neurons are both active when preparing and when observing the action.

After the discovery of mirror neurons in monkeys it has been hypothesized that similar types of neurons also occur in humans. Indeed, for humans from the usual imaging methods it can be found that in certain premotor areas activity occurs both when an action is observed and when the action is prepared; e.g., [11], [25] based on EEG data; [27], [49] based on PET data, [36] based on fMRI. However, due to limitations in resolution, from such methods it cannot be seen whether the neurons active in action observation are exactly the same neurons that are active in preparing for an action. In principle they could be different neurons in the same area, each with only one function: either observation or preparing. Therefore in the years after the discovery of mirror neurons in monkeys it has been subject to debate whether they also exist in humans; e.g., [31].

Recently the existence of mirror neurons in humans has found support in single cell experiments with epilepsy patients undergoing pre-surgical evaluation of the foci of epilepsy; cf. [21], [43]; see also [34], pp. 201-203; [35], [38]. In these experiments for 14 patients, the activity of approximately 500 neurons was recorded; they were located in three sectors of the mesial frontal cortex (the ventral and dorsal sectors of the anterior cingulate cortex and the pre-supplementary motor cortex (SMA)/SMA proper complex). The subjects were tested both for hand-grasping actions and for emotional face expressions. Some of the main findings were that neurons with mirror

neuron properties were found in all sites in the mesial frontal cortex where recording took place, in total for approximately 12% of all recorded neurons; about half of them related to hand-grasping, and the other half to emotional face expressions; cf. [35].

2.2 Super Mirror Neurons for Control and Self-other Distinction

Due to the multiple functions of mirror neurons, the functional meaning of activation of them (e.g., preparing or observing an action, or both) in principle is context-dependent. The context determines in which cases their activation is meant to lead to actual execution of the action (e.g., in self-initiated action performance, or imitation), and in which cases it is not (e.g., in action observation). A specific subset of mirror neurons has been found, called *super mirror neurons* that seem to be able to indicate such a context and play a role in the control of actual execution of a prepared action. These neurons are suggested to exert control by allowing or suppressing action execution, and/or by suppressing preparation states. More details on super mirror neurons can be found in [7], [35], and [34], pp. 196-203.

In the single cell recording experiments with epileptic patients mentioned above, also cells were found that are active when the person prepares an own action that is executed, but shut down when the action is only observed. This has led to the hypothesis that these cells may be involved in the functional distinction between a preparation state activated in order to actually perform the action, a preparation state activated to interpret an observed action (or both, in case of imitation). In [34], pp. 201-202 it is also described that some of such cells are sensitive to a specific person, so that the action can be attributed to the specific person that was observed: self-other distinction; see also [7].

2.3 Mirroring: Mirror Neuron Activation and Internal Simulation

Activation states of mirror neurons are important not by themselves, but because they play a crucial role in an important mental function: *mirroring* mental processes of other persons by *internal simulation*. How mirroring relates to internal processes involving emotions and feelings may ask for some further explanation. A classical view on emotions is that based on some sensory input due to internal processing emotions are felt, and based on that they are expressed in some body state; e.g., a face expression:

sensory representation → felt emotion → preparation for bodily changes →
expressed bodily changes = expressed emotion

James [37] claimed a different direction of causality (see also [17], pp. 114-116):

sensory representation → preparation for bodily changes → expressed bodily changes
→ emotion felt = based on sensory representation of (sensed) bodily changes

The perspective of James assumes that a *body loop* is used to generate an emotion. Damasio made an important further step by introducing the possibility of an *as-if body loop* bypassing (the need for) actually expressed bodily changes (cf. [13], pp. 155-158; see also [15], pp. 79-80; [16], [17]):

sensory representation → preparation for bodily changes = emotional response →
emotion felt = based on sensory representation of (simulated) bodily changes

An as-if body loop describes an *internal simulation* of the bodily processes, without actually affecting the body, comparable to simulation in order to perform, for example, prediction, mindreading or imagination; e.g., [2], [24], [26], [30]; see also [6], [29] for computational accounts. The feelings generated in this way play an important role in valuing predicted or imagined effects of actions (in relation to amygdala activations; see, e.g., [42], [44]). Note that, in contrast to James [37], Damasio [13] distinguishes an emotion (or emotional response) from a feeling (or felt emotion). The emotion and feeling in principle mutually affect each other in a bidirectional manner: an as-if body loop usually occurs in a cyclic form by assuming that the emotion felt in turn affects the prepared bodily changes; see, for example, in ([16], pp. 91-92; [17], pp. 119-122):

emotion felt = based on sensory representation of (simulated) bodily changes →
preparation for bodily changes = emotional response

This provides a cyclic process that (for a constant environment) can lead to equilibrium states for both, as shown, for example, in [6] by a computational model. From a more general viewpoint, as-if body loops as introduced in [13] contribute:

- (1) sensory input directly affects preparation states, after which further internal processing takes place (in line with, e.g., [37])
- (2) the notion of internal simulation (in line with, e.g., [2], [30]).

Here (1) breaks with the tradition that there is a standard order of processing sensing – internal processing – preparation for action, and (2) allows for involving body representations in internal processes without actually having to change any body state. As mirror neurons make that some specific sensory input (an observed person) directly links to related preparation states, just like (1) above, it fits quite well in the perspective based on as-if body loops. In this way mirroring is a process that fully integrates mirror neuron activation states in the ongoing internal processes based on as-if loops; see also [17], pp. 102-104. As this happens mostly in an unconscious manner, mirroring imposes limitations on the freedom for individuals to have their own personal emotions, beliefs, intentions, and actions.

2.4 Development of the Discipline Social Neuroscience

Above it has been pointed out how states of other persons lead to activation of some of a person's corresponding own states that at the same time play a crucial role in the person's feelings and actions. This provides an effective mechanism for how observed actions and feelings and own actions and feelings are tuned to each other. This mechanism explains how in a social context persons fundamentally affect each other's personal actions and states, including feelings. Given these implications, the discovery of mirror neurons and how they play their role in mirroring processes is considered a crucial step for the further development of the disciplines of social cognition and social psychology, by providing a biological basis for many social phenomena.

Many examples of social phenomena now can be related to mirroring, for example: social diffusion or contagion of personal states such as opinions or emotions; empathic understanding; group formation, group cohesion, collective decision making. Based on these developments, and their wide applicability the new discipline Social Neuroscience has shown a fast development; e.g., [9], [10], [19], [20], [28]. The impact of this discipline is very wide, as it is considered to cover the concept of social reality (e.g., [8]), spiritual and religious experience (e.g., [52]), and collective consciousness or global empathy and its role in the future evolution (e.g., [12], [47]). In the next two sections it will be discussed how different types of shared understanding and collective power can emerge based on the mirroring function.

3 The Emergence of Shared Understanding

Understanding can occur in different forms. An agent can have an understanding of a world state by generating and maintaining an internal cognitive state in relation to it (e.g., one or more beliefs about it). This can be distinguished as a *cognitive* type of *understanding*. An agent can also form and maintain an internal affective state in relation to a world state (e.g., a specific emotion or feeling associated to it). Such a form of understanding can be distinguished as an *affective* type of *understanding*. An important role of this type of understanding is that it provides a basis for *experiencing* in the understanding. Affective and cognitive understanding are often related to each other. Any cognitive state triggers an associated emotional response which based on an as-if body loop activates a sensory representation of a body state which is the basis of the related feeling (e.g., [13], [15], [16], [17]); see also Section 2. Assuming similar neural architectures, the associated emotion is generated in an observing agent just like it is in an observed agent. In this way, mirroring is a mechanism to obtain shared understanding integrating cognitive and affective aspects.

A second way of distinguishing different types of understanding is by considering that the concerning world state can be either an *agent-external* world state or an *agent-internal* world state. For example, having beliefs about another agent's emotions, beliefs or goals is of the second, agent-internal type, whereas having beliefs about the weather is of the first type. These two dimensions of distinctions introduced above can be applied to *shared* understanding of an agent B with an agent A, from which a matrix results as illustrated in Table 1 with different examples.

Table 1. Examples of different types of shared understanding

	<i>Agent-internal</i>	<i>Agent-external</i>
<i>Shared cognitive understanding</i>	<ul style="list-style-type: none"> • having beliefs about agent A's beliefs, intentions or goals • sharing goals for an internal agent state 	<ul style="list-style-type: none"> • sharing beliefs with agent A about an external world state • sharing goals for an external world state
<i>Shared affective understanding</i>	<ul style="list-style-type: none"> • feeling the same as agent A is feeling about an agent state 	<ul style="list-style-type: none"> • sharing a good or bad feeling about an external world state
<i>Shared cognitive and affective understanding</i>	<ul style="list-style-type: none"> • believing that agent A feels bad • believing X and feeling Y, and believing that agent A also believes X and feels Y 	<ul style="list-style-type: none"> • sharing a belief or goal and feeling • sharing a belief and a feeling that intention X will achieve goal Y

3.1 The Emergence of Shared Understanding for Agent-External States

An agent's understanding of the external world in the form of a collection of beliefs is sometimes called the agent's world model. This can be considered a cognitive world model. More general, shared understanding of an external world state can involve:

- a *shared cognitive world model* (e.g., sharing beliefs about an external world state)
- a *shared affective world model* (e.g., sharing feelings about an external world state)
- a combined *shared cognitive-affective world model* (e.g., sharing both)

An example of the last item is sharing a belief that climate change has some serious effects and sharing a bad feeling about that, or sharing a belief that a new iPad will come out soon and sharing a good feeling about that. Obtaining such shared understanding of the external world may make use of different means. Individual information gathering can play a role, but also verbal and nonverbal interaction between agents. If some external world state is considered by agents, both verbal and nonverbal expressions are input for mirroring processes. These mirroring processes affect, for example, both the strength by which something is believed about this state, and the strength of the feeling associated to it. Thus both cognitive and affective shared understanding can develop, based on (mostly unconscious) mirroring processes.

3.2 The Emergence of Shared Understanding for Agent-Internal States

A second type of understanding concerns states that are internal for one of the agents. For such understanding different terms are used; e.g., mindreading, Theory of Mind (ToM), empathy, or more specific terms such as emotion or intention recognition; e.g., [20], [26], [46]. Also here understanding may be limited to cognitive understanding; for example, believing that another person has the intention to go out for a dinner, or feels depressed. However, for humans also an affective type of mutual understanding is common, usually combined with some form of cognitive understanding. One of the most fundamental forms of mutual understanding is indicated by the notion of *empathy*; e.g., see [18], [20], [34], [46], [53], [54], [57]. Originally by Lipps [40] the notion was named by the German word 'einfühlung' which could be translated as 'feeling into'; e.g., [46]. As this word indicates more explicitly, the notion of empathy has a strong relation to feeling: *empathic understanding* includes experiencing what the other person feels, but also believing that the experienced feeling is felt by the other person, based on self-other distinction (a form of super mirroring). Therefore empathic understanding can be considered a form of combined affective and cognitive understanding; see also [53], [54]. As an example, in [57], and [18], p. 435, the following four criteria of empathy of *B* for *A* are formulated:

- (1) Presence of an affective state in a person *B*
- (2) Isomorphism of *B*'s own and *A*'s affective state
- (3) Elicitation of the *B*'s affective state upon observation or imagination of *A*'s affective state
- (4) Knowledge of *B* that *A*'s affective state is the source of the *B*'s own affective state

The understanding indeed is both affective (1) and cognitive (4), but in this case it concerns in particular an affective state and not a cognitive state of the other person. Therefore it can be called *affective-focused empathy*. In contrast, to indicate affective and cognitive understanding of another agent's cognitive state (e.g., a belief) the term *cognitive-focused empathy* may be used. The term (*full*) *empathy* can be used to indicate combined cognitive-affective understanding of both cognitive and (associated) affective states of another agent. Note that empathy always involves feelings, so this is also the case, for example, in cognitive-focused empathy. However, in case of full empathy these feelings are related to the other person (using self-other distinction), and in case of purely cognitive-focused empathy the feelings are experienced but not related to the other person (for example, due to impaired self-other distinction). Table 2 illustrates these types of understanding for agent *B* having understanding of states of agent *A*. That mirroring (together with super mirroring) provides a basic mechanism involved in the creation of empathic understanding has much support in the recent literature; e.g., [22], [53], [54], [57], and [34], pp. 106-129.

Table 2. Examples of different types of Theory of Mind and empathy of agent *B* w.r.t. agent *A*

Agent A Agent B	<i>Affective states</i>	<i>Cognitive states</i>	<i>Affective and cognitive states</i>
<i>Affective understanding</i>	Feeling but not having a belief for <i>A</i> 's emotion (<i>emotion contagion</i>)	Feeling but not having a belief for <i>A</i> 's belief	Feeling but not having a belief for <i>A</i> 's emotion and belief
<i>Cognitive understanding</i>	Having a belief but no feeling for <i>A</i> 's emotion (<i>affective-focused ToM</i>)	Having a belief but no feeling for <i>A</i> 's belief (<i>cognitive-focused ToM</i>)	Having a belief but no feeling for <i>A</i> 's emotion and belief (<i>ToM</i>)
<i>Affective and cognitive understanding</i>	Having both a belief and feeling for <i>A</i> 's emotion (<i>affective-focused empathy</i>)	Having both a belief and feeling for <i>A</i> 's belief (<i>cognitive-focused empathy</i>)	Having a belief and feeling for <i>A</i> 's belief and feeling (<i>empathy</i>)

4 The Emergence of Collective Power

Each individual can exert a certain amount and direction of power by his or her actions, depending on personal characteristics and states. In a situation where such powers are exerted in different directions by multiple individuals, they can easily annihilate each other, or result in a kind of Brownian motion where particles move back and forth but do not change place much. In cases that the individual momenta represented by the individual powers and their directions, have an arbitrary distribution over a population, no serious collective momentum will emerge.

4.1 The Emergence of Collective Action Based on Mirroring

To obtain emergence of collective power, the individual momenta should converge to a similar direction so that a collective momentum can result. To obtain this, within

groups of agents, shared agent states can emerge (by mirroring) that in an anticipatory sense relate to action, and by which collective power can be developed. Types of internal states relating to action are intentions or preparations. They can be seen as tendencies to perform a specific action; the emergence of shared preparations by mirroring may be quite effective in this sense. Such a process may play an important role in emerging collective decision making: a mirroring process may achieve that a specific preparation option gets a high activation level for all individuals in a group.

4.2 The Role of Feelings and Valuing in the Emergence of Collective Action

Usually in the individual process of action selection, before a prepared action comes in focus to be executed, an internal simulation to predict the effects of the action takes place: the action is simulated based on prediction links, and in particular for the associated affective effects, based on as-if body loops that predict the body state which is the basis of the related feeling (e.g., [13], [15], [16], [17]). Based on these predicted effects a valuation of the action takes place, which may involve or even be mainly based on the associated affective state, as, for example, described in [1], [13], [14], [16], [42], [44]. The idea here is that by an as-if body loop each option (prepared action) induces a simulated effect including a feeling which is used to value the option. For example, when a negative feeling and value is induced by a particular option, it provides a negative assessment of that option, whereas a positive feeling and value provides a positive assessment. The decision for executing a prepared action is based on the most positive assessment for it.

This process of simulating effects of prepared actions not only takes place for preparations of self-generated actions, but also for intentions or actions from other persons that are observed. In this way by the mirroring process not only a form of action or intention recognition takes place in the form of activation of corresponding own preparation states by mirror neurons, but in addition also the (predicted) effects are simulated, including the affective effects. This provides an emotionally grounded form of understanding of the observed intention or action, including its valuing, which is shared with the observed agent; see also [17], pp. 102-104.

Given the important role of the feeling states associated to preparations of actions, it may be unrealistic to expect that a common action can be strong when the individual feelings and valuations about such an action have much variation over a group. To achieve emergence of strong collective action, also a shared feeling and valuation for this action has to develop: also mirroring of the associated emotions has to play an important role. When this is achieved, the collective action has a solid shared emotional grounding: the group members do not only intend to perform that action collectively, but they also share a good feeling about it. In this process social media can play an important facilitating role in that (1) they dramatically strengthen the connections between large numbers of individuals, and (2) they do not only support transfer of, for example, beliefs and intentions as such, but also associated emotions reinforcing them, thus making the transfer double effective.

5 Computational Models for Social Agents

In this section, the processes discussed above are illustrated by (pointers to) examples of computational models. For example, in [6] and [39] it is shown how mirroring plays a role in emotion recognition. Examples with both mirroring and super mirroring functions can be found in [29], [58], [59]. In [29] it is shown how depending on the context represented by super mirror states, activation of a preparation state has a function in either execution, recognition, imagination or imitation of an action. In [58] it is shown how super mirror states play a role in regulation of different forms of social response patterns, and in [59] in prior and retrospective ownership states for an action.

Computational models for the emergence of shared understanding of agent-external states can be found, for example, in [4] where a computational model for converging emotion spirals (e.g., of fear) is described. In [32] a computational model for cognitive states (beliefs), and affective states (fear) with respect to the external world (in mutual relation) is described which shows how for such combined cases shared understanding emerges. Computational models that have been developed for different types of shared understanding of agent-internal states based on a mirroring mechanism, can be found, for example, in [6] and [58] for affective-focused empathic understanding and social responses, and in [41] for full empathic understanding.

5.1 A Generic Contagion Model

As a further illustration, first a model is briefly described where a person's internal states are fully determined by other persons' states, and not by other internal processes (taken from [4]). This model describes at an abstract level the mirroring of any given mental state S (for example, an emotion or intention). An important element is the contagion strength γ_{SBA} for S from person B to person A . This indicates the strength by which the state S of A is affected by the state S of B . It depends on characteristics of the two persons: how expressive B is, how open A is, and how strong the connection from B to A is. In the model it is defined by

$$\gamma_{SBA} = \varepsilon_{SB} \alpha_{SBA} \delta_{SA}.$$

Here, ε_{SB} is the *expressiveness* of B for S , δ_{SA} the *openness* of A for S , and α_{SBA} the *channel strength* for S from B to A . The level q_{SA} of state S in agent A (with values in the interval $[0, 1]$) over time is determined as follows. The overall contagion strength γ_{SA} from the rest of the group towards agent A is $\gamma_{SA} = \sum_{B \neq A} \gamma_{SBA}$. The aggregated impact q_{SA}^* of all these agents upon state S of agent A is:

$$q_{SA}^*(t) = \sum_{B \neq A} \gamma_{SBA} q_{SB}(t) / \gamma_{SA}$$

Given these, the dynamics of the level of state S in A are modelled by:

$$q_{SA}(t+\Delta t) = q_{SA}(t) + \gamma_{SA} [f(q_{SA}^*(t), q_{SA}(t)) - q_{SA}(t)] \Delta t$$

where $f(X, Y)$ is a combination function, which can be taken, for example, as:

$$\begin{aligned} f(X, Y) &= \alpha X + (1-\alpha)Y && \text{(absorption, no bias)} \\ f(X, Y) &= \beta_{SA} (1 - (1-X)(1-Y)) + (1-\beta_{SA}) XY && \text{(amplification, bias } \beta_{SA}) \end{aligned}$$

The parameter β_{SA} with values between 0 and 1 indicates a bias towards increasing (upward, $\beta_{SA} > 0.5$) or reducing (downward, $\beta_{SA} < 0.5$) the impact for the value of the state S of A . Some example simulations for levels of an emotion state S using the latter combination function are shown in Fig. 1 for three agents a, b, c (taken from [4]). When there are no biases (i.e., all $\beta_{SA} = 0.5$), then a shared level emerges which is a weighted average of the individual initial values; an example of this is shown in Fig. 1(a). The way in which these initial values are weighted depends on the openness, expressiveness and channel parameters. If one of the group members is a charismatic leader figure, with very high expressiveness ε and very low openness δ , then this person's initial state will dominate in the emerging shared state, for example, as thought for persons like Lech Wałęsa, Winston Churchill, Adolph Hitler, Martin Luther King Jr, Fidel Castro, Mahatma Gandhi. Persons with high openness and low expressivity are considered to be followers, persons with low openness and expressiveness loners. Social media can play an important role because they increase the channel strengths α between individuals for, for example, beliefs and intentions, and also for the associated emotions. In Fig. 1(b) a situation is shown where biases play a role; here the emerging shared emotion level is higher than any of the initial individual values. Note that the bias of agent b is downward (value $0.3 < 0.5$), which indeed for this agent leads to a downward trend first; this trend is changing after time point 2, due to the impact of other, as by then the other agents' emotion levels have substantially increased.

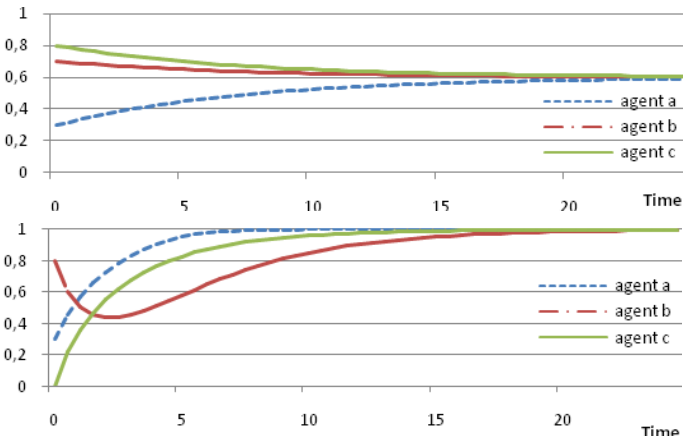


Fig. 1. Emerging shared emotion states depending on bias values (a) Not biased: all $\beta_A = 0.5$ (b) Biased: $\beta_a = 1, \beta_b = 0.3, \beta_c = 0.8$

5.2 Integration of Mirroring in other Internal Processes

The generic model described above applies to any internal state S , but does not describe any interplay between different internal states yet. In more realistic cases such an interplay also contributes to the levels of the states, and therefore the impact of other internal states S' on a given state S has to be integrated with the impact of

mirroring on S . For example, Fig. 2 shows an internal model where a certain cognitive state (for example, a sensory representation or belief) has both a cognitive and affective impact on a person's emotions and preparations. Usually such impacts also have feedback loops; an example of this is an as-if body loop (see Section 2). Therefore, often an internal model consists of a number of cycles, for example, as shown in Fig. 2. In processing, these loops may converge to some equilibrium, when impact from outside is not changing too fast.

An integration of such an internal model with external impact by mirroring can be obtained as follows: to update $q_{SA}(t)$ for a state S , the levels $q_{S'A}(t)$ for the other states S' are taken into account. A general way to do this is by a combination function f that both takes into account the aggregated mirroring impact $q_{SA}^*(t)$ and the values $q_{S'A}(t)$ for all (relevant) internal states $S' \neq S$. A relatively simple way to define such a combination function is by a weighted average of all these impacts:

$$q_{SA}^{**}(t) = \lambda_{SA} q_{SA}^*(t) + \sum_{S' \neq S} \lambda_{S'A} q_{S'A}(t) \quad \text{with } \sum_{S'} \lambda_{S'A} = 1$$

and then in the dynamic update model for $q_{SA}(t)$ described above in the combination function $f(X, Y)$ use this $q_{SA}^{**}(t)$ instead of $q_{SA}^*(t)$. This way of combination was used in the computational model for emotion-grounded collective decision making described in [32], based on the principles discussed in Section 4 above. In this case mirroring was applied to both emotion and intention states for any option O :

- *mirroring of emotions* as a mechanism for how emotions felt about a certain considered decision option O in different individuals mutually affect each other
- *mirroring of intentions* as a mechanism for how strengths of intentions (action tendencies) for a certain decision option O in different individuals affect each other

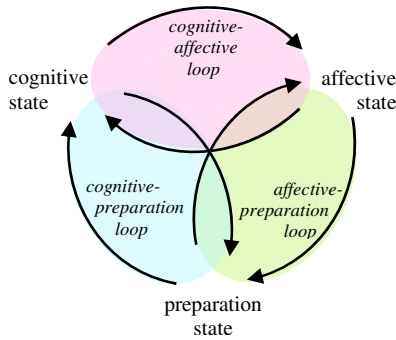


Fig. 2. Example of a more complex internal model

In the model not only intentions of others, but also a person's emotions affect the person's own intentions (the arrow from affective state to preparation state in Fig. 2). In updating $q_{SA}(t)$ for an intention state S relating to an option O , the intention states of others for O and the values for the emotion state S' for O were taken into account, and aggregated using the approach indicated above. In simulations in most cases not only a collective decision for an intention was emerging, but also a shared underlying feeling. For more details and simulation results, see [32]. Examples of exceptions occur when group members have no openness for others, or are not connected to others.

5.3 The Interplay of Intentions, Beliefs and Emotions

An example of a more complex computational model is the collective decision making model ASCRIBE addressing an interplay of beliefs, intentions, and emotions (Agent-based Social Contagion Regarding Intentions, Beliefs and Emotions; cf. [33]); see Fig. 3. The internal model used here instantiates part of the general picture of Fig. 2. Beliefs instantiate the cognitive, emotions the affective, and intentions the preparation states. In this specific internal model it is assumed that an individual’s strength of an intention for a certain decision option depends on the person’s beliefs (*cognitive responding*) and emotions (*somatic marking*) in relation to that option. Moreover, it is assumed that beliefs may generate certain emotions (*affective responding*), for example of fear, that in turn may affect the strength of beliefs (*affective biasing*). Note that these latter emotion impacts are independent of specific decision options (e.g., a general fear level). Mirroring was used in three different forms (the dotted arrows in Fig. 3): of emotions (both fear and emotions felt about a certain decision option O), of beliefs, and of intentions (for a certain decision option O). In the model for the dynamics of intentions, the impact from mirroring is combined with impact from the emotion states and impact from beliefs, in a similar manner as described above. The same applies, for example, to the impact of beliefs on the emotion state. However, in this model also a different type of combination of mirroring and internal processes takes place, involving impact of fear states to beliefs: it is assumed that some of the parameters, for example, for biases and openness with respect to beliefs are affected by fear levels. For more details of this model and example simulations, see [33]; in [5] an application to a real world crowd behaviour case is presented.

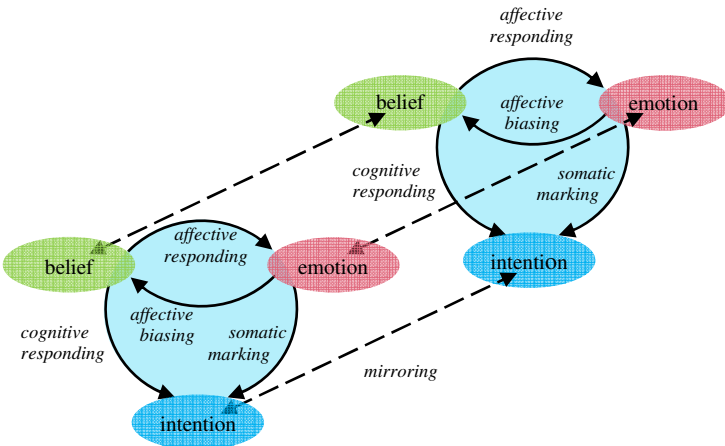


Fig. 3. Threefold mirroring integrated with internal interplay of beliefs, emotions and intentions

6 Discussion

In this paper it was discussed how mechanisms from the new discipline Social Neuroscience can be exploited to obtain social agent models, covering both cognitive and affective processes, and their interaction. Core mechanisms used are mirror neurons and internal simulation. Mirror neurons are certain neurons that are activated due to observation of another agent having a corresponding state; e.g., [34], [45], [51]. Internal simulation is further internal processing copying a process within another person; e.g., [13], [15], [24], [26], [30]. It was shown how such agent models can be used to perform simulation and analysis of the emergence of shared understanding of a group of agents. Furthermore, it was shown how such agent models can be used to perform simulation and analysis of the emergence of collective power of a group of agents. This was addressed both in a cognitive or affective or combined sense, so that not only the group members together go for a collective action, but they also share the experience of a good feeling about it, which gives the collective action a solid emotional grounding. It was discussed how such processes depend on the connection strengths between persons, which are strengthened, for example, by social media.

The obtained agent models were specified as internal models at the cognitive and affective level, and often involve loops between different internal states. However, under certain assumptions such internal models can be abstracted to behavioural model providing more efficient processing, which is important especially when larger numbers of agents are simulated; for more details; for example, see [55], [56].

The perspective put forward in this paper has a number of possible application areas. In the first place it can be used to analyse human social processes in groups, crowds or in societies as a whole. The application to crowd behaviour in emergency situations addressed in [5] is an example of such an application. Other cases address, for example, collective decision making, the construction of social reality (e.g., [8]), the development of collective consciousness (e.g., [12]), and global empathy enabling to solve global problems such as climate change (e.g., [47]), or spiritual and religious experience (e.g., [52]).

A second area of application addresses groups of agents that partly consist of human agents and partly of devices, such as smartphones, and use of social media. For such mixed groups it can not only be analysed what patterns may emerge, but also the design of these devices and media can be an aim, in order to create a situation that the right types of patterns emerge, for example, with safe evacuation as a consequence.

A third area of application concerns a close empathic interaction between a human and a device. The importance of computational models in a virtual context for 'caring' agents showing empathy has also been well-recognized in the literature; see, for example [3]. As a fourth area of application team formation can be addressed. In this area it may be analysed in what way the above perspective provides possibilities that differ compared to already existing approaches.

References

1. Bechara, A., Damasio, H., Damasio, A.R.: Role of the Amygdala in Decision-Making. *Ann. N.Y. Acad. Sci.* 985, 356–369 (2003)
2. Becker, W., Fuchs, A.F.: Prediction in the Oculomotor System: Smooth Pursuit During Transient Disappearance of a Visual Target. *Experimental Brain Research* 57, 562–575 (1985)
3. Bickmore, T.W., Picard, R.W.: Towards Caring Machines. In: Dykstra-Erickson, E., Tscheligi, M. (eds.) *Proceedings of CHI 2004*, pp. 1489–1492. ACM, New York (2004)
4. Bosse, T., Duell, R., Memon, Z.A., Treur, J., van der Wal, C.N.: A Multi-agent Model for Emotion Contagion Spirals Integrated within a Supporting Ambient Agent Model. In: Yang, J.-J., Yokoo, M., Ito, T., Jin, Z., Scerri, P. (eds.) *PRIMA 2009. LNCS(LNAI)*, vol. 5925, pp. 48–67. Springer, Heidelberg (2009)
5. Bosse, T., Hoogendoorn, M., Klein, M.C.A., Treur, J., van der Wal, C.N.: Agent-Based Analysis of Patterns in Crowd Behaviour Involving Contagion of Mental States. In: Mehrotra, K.G., Mohan, C.K., Oh, J.C., Varshney, P.K., Ali, M. (eds.) *IEA/AIE 2011, Part II. LNCS(LNAI)*, vol. 6704, pp. 566–577. Springer, Heidelberg (2011)
6. Bosse, T., Memon, Z.A., Treur, J.: A Cognitive and Neural Model for Adaptive Emotion Reading by Mirroring Preparation States and Hebbian Learning. *Cognitive Systems Research* (2011) (in press),
<http://dx.doi.org/10.1016/j.cogsys.2010.10.003>
7. Brass, M., Spengler, S.: The Inhibition of Imitative Behaviour and Attribution of Mental States. In: Striano, T., Reid, V. (eds.) *Social Cognition: Development, Neuroscience, and Autism*, pp. 52–66. Wiley-Blackwell (2009)
8. Butz, M.V.: Intentions and Mirror Neurons: From the Individual to Overall Social Reality. *Constructivist Foundations* 3, 87–89 (2008)
9. Cacioppo, J.T., Berntson, G.G.: *Social neuroscience*. Psychology Press, San Diego (2005)
10. Cacioppo, J.T., Visser, P.S., Pickett, C.L.: *Social neuroscience: People thinking about thinking people*. MIT Press, Cambridge (2006)
11. Cochin, S., Barthelemy, B., Roux, S., Martineau, J.: Observation and Execution of movement similarities demonstrated by quantified electroencephalography. *European Journal of Neuroscience* 11, 1839–1842 (1999)
12. Combs, A., Krippner, S.: Collective Consciousness and the Social Brain. *Journal of Consciousness Studies* 15, 264–276 (2008)
13. Damasio, A.: *Descartes' Error: Emotion, Reason and the Human Brain*. Papermac, London (1994)
14. Damasio, A.: The Somatic Marker Hypothesis and the Possible Functions of the Prefrontal Cortex. *Philosophical Transactions of the Royal Society: Biological Sciences* 351, 1413–1420 (1996)
15. Damasio, A.: *The Feeling of What Happens. Body and Emotion in the Making of Consciousness*. Harcourt Brace, New York (1999)
16. Damasio, A.: *Looking for Spinoza: Joy, Sorrow, and the Feeling Brain*. Vintage books, London (2003)
17. Damasio, A.R.: *Self comes to mind: constructing the conscious brain*. Pantheon Books, NY (2010)
18. De Vignemont, F., Singer, T.: The empathic brain: how, when and why? *Trends in Cogn. Sciences* 10, 437–443 (2006)
19. Decety, J., Cacioppo, J.T. (eds.): *Handbook of Social Neuroscience*. Oxford University Press, Oxford (2010)

20. Decety, J., Ickes, W.: *The Social Neuroscience of Empathy*. MIT Press, Cambridge (2009)
21. Fried, I., Mukamel, R., Kreiman, G.: Internally Generated Preactivation of Single Neurons in Human Medial Frontal Cortex Predicts Volition. *Neuron* 69, 548–562 (2011)
22. Gallese, V.: The Roots of Empathy: The Shared Manifold Hypothesis and the Neural Basis of Intersubjectivity. *Psychopathology* 36, 171–180 (2003)
23. Gallese, V., Fadiga, L., Fogassi, L., Rizzolatti, G.: Action Recognition in the Premotor Cortex. *Brain* 119, 593–609 (1996)
24. Gallese, V., Goldman, A.: Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences* 2, 493–501 (1998)
25. Gastout, H.J., Bert, J.: EEG changes during cinematographic presentation. *Electroencephalography and Clinical Neurophysiology* 6, 433–444 (1954)
26. Goldman, A.I.: *Simulating Minds: The Philosophy, Psychology, and Neuroscience of Mindreading*. Oxford Univ. Press, New York (2006)
27. Grafton, S.T., Arbib, M.A., Fadiga, L., Rizzolatti, G.: Localisation of grasp representations in humans by PET: 2. Observation Compared with Imagination. *Experimental Brain Research* 112, 103–111 (1996)
28. Harmon-Jones, E., Winkielman, P. (eds.): *Social neuroscience: Integrating biological and psychological explanations of social behavior*. Guilford, New York (2007)
29. Hendriks, M., Treur, J.: Modeling Super Mirroring Functionality in Action Execution, Imagination, Mirroring, and Imitation. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) *ICCCI 2010. LNCS*, vol. 6421, pp. 330–342. Springer, Heidelberg (2010)
30. Hesslow, G.: Conscious thought as simulation of behaviour and perception. *Trends Cogn. Sci.* 6, 242–247 (2002)
31. Hickok, G.: Eight Problems for the Mirror Neuron Theory of Action Understanding in Monkeys and Humans. *Journal of Cognitive Neuroscience* 21, 1229–1243 (2009)
32. Hoogendoorn, M., Treur, J., van der Wal, C.N., van Wissen, A.: Agent-Based Modelling of the Emergence of Collective States Based on Contagion of Individual States in Groups. *Transactions on Computational Collective Intelligence* 3, 152–179 (2011)
33. Hoogendoorn, M., Treur, J., van der Wal, C.N., van Wissen, A.: Modelling the Interplay of Emotions, Beliefs and Intentions within Collective Decision Making Based on Insights from Social Neuroscience. In: Wong, L.K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) *ICONIP 2010, Part I. LNCS*, vol. 6443, pp. 196–206. Springer, Heidelberg (2010)
34. Iacoboni, M.: *Mirroring People: the New Science of How We Connect with Others*. Farrar, Straus & Giroux, New York (2008)
35. Iacoboni, M.: Mesial frontal cortex and super mirror neurons. *Behavioral and Brain Sciences* 31, 30–30 (2008)
36. Iacoboni, M., Molnar-Szakacs, I., Gallese, V., Buccino, G., Mazziotta, J.C., Rizzolatti, G.: Grasping the intentions of others with one's own mirror neuron system. *PLoS Biology* 3, e79 (2005)
37. James, W.: What is an emotion. *Mind* 9, 188–205 (1884)
38. Keysers, C., Gazzola, V.: Social Neuroscience: Mirror Neurons Recorded in Humans. *Current Biology* 20, 253–254 (2010)
39. van der Laan, Y., Treur, J.: An Agent Model for Computational Analysis of Mirroring Dysfunctioning in Autism Spectrum Disorders. In: Mehrotra, K.G., Mohan, C.K., Oh, J.C., Varshney, P.K., Ali, M. (eds.) *IEA/AIE 2011, Part I. LNCS(LNAI)*, vol. 6703, pp. 306–316. Springer, Heidelberg (2011)
40. Lipps, T.: Einfühlung, innere Nachahmung und Organempfindung. *Archiv für die Gesamte Psychologie* 1, 465–519 (1903)

41. Memon, Z.A., Treur, J.: An Agent Model for Cognitive and Affective Empathic Understanding of Other Agents. In: Transactions on Computational Collective Intelligence (to appear, 2011); earlier, shorter version in: Nguyen, N.T., Kowalczyk, R., Chen, S.M. (eds.): ICCCI 2009. LNCS, vol. 5796, pp. 279–293. Springer, Heidelberg (2009)
42. Morrison, S.E., Salzman, C.D.: Re-valuing the amygdala. *Current Opinion in Neurobiology* 20, 221–230 (2010)
43. Mukamel, R., Ekstrom, A.D., Kaplan, J., Iacoboni, M., Fried, I.: Single-Neuron Responses in Humans during Execution and Observation of Actions. *Current Biology* 20, 750–756 (2010)
44. Murray, E.A.: The amygdala, reward and emotion. *Trends Cogn. Sci.* 11, 489–497 (2007)
45. Pineda, J.A. (ed.): *Mirror Neuron Systems: the Role of Mirroring Processes in Social Cognition*. Humana Press Inc., Totowa (2009)
46. Preston, S.D., de Waal, F.B.M.: Empathy: its ultimate and proximate bases. *Behav. Brain Sci.* 25, 1–72 (2002)
47. Rifkin, J.: *The Empathic Civilization: The Race to Global Consciousness in a World in Crisis*. Tarcher Penguin (2010)
48. Rizzolatti, G., Fadiga, L., Gallese, V., Fogassi, L.: Premotor Cortex and the Recognition of Motor Actions. *Cognitive Brain Research* 3, 131–141 (1996)
49. Rizzolatti, G., Fogassi, L., Matelli, M., et al.: Localisation of grasp representations in humans by PET: 1. Observation and Execution. *Experimental Brain Research* 111, 246–252 (1996)
50. Rizzolatti, G., Craighero, L.: The Mirror Neuron System. *Annual Review of Neuroscience* 27, 169–192 (2004)
51. Rizzolatti, G., Sinigaglia, C.: *Mirrors in the Brain: How Our Minds Share Actions and Emotions*. Oxford University Press, Oxford (2008)
52. Seybold, K.S.: Biology of Spirituality. *Perspectives on Science and Christian Faith* 62, 89–98 (2010)
53. Shamay-Tsoory, S.G.: Empathic processing: its cognitive and affective dimensions and neuroanatomical basis. In: Decety, J., Ickes, W. (eds.) *The Social Neuroscience of Empathy*, pp. 215–232. MIT Press, Cambridge (2008)
54. Shamay-Tsoory, S.G.: The Neural Bases for Empathy. *Neurosc.* 17, 18–24 (2011)
55. Sharpanskykh, A., Treur, J.: Abstraction Relations between Internal and Behavioural Agent Models for Collective Decision Making. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010. LNCS(LNAI), vol. 6421, pp. 39–53. Springer, Heidelberg (2010); Extended version in: *Web Intelligence and Agent Systems* (to appear, 2011)
56. Sharpanskykh, A., Treur, J.: Behavioural Abstraction of Agent Models Addressing Mutual Interaction of Cognitive and Affective Processes. In: Yao, Y., Sun, R., Poggio, T., Liu, J., Zhong, N., Huang, J. (eds.) BI 2010. LNCS(LNAI), vol. 6334, pp. 67–77. Springer, Heidelberg (2010)
57. Singer, T., Leiberg, S.: Sharing the Emotions of Others: The Neural Bases of Empathy. In: Gazzaniga, M.S. (ed.) *The Cognitive Neurosciences*, 4th edn., pp. 973–986. MIT Press, Cambridge (2009)
58. Treur, J.: A Cognitive Agent Model Displaying and Regulating Different Social Response Patterns. In: Walsh, T. (ed.) *Proc. IJCAI 2011*, pp. 1735–1742 (2011)
59. Treur, J.: A Cognitive Agent Model Incorporating Prior and Retrospective Ownership States for Actions. In: Walsh, T. (ed.) *Proc. IJCAI 2011*, pp. 1743–1749 (2011)

Experiential Knowledge in the Development of Decisional DNA (DDNA) and Decisional Trust for Global e-Decisional Community

Edward Szczerbicki and Cesar Sanin

University of Newcastle, Newcastle, Australia

Abstract. In the nineties, Peter Drucker envisaged that “the traditional factors of production – land, labour and capital are becoming restraints rather than driving forces” and “Knowledge is becoming the one critical factor of production”. Welcoming the onset of knowledge society, we are proposing in this paper an approach the aim of which is to develop a concept, tools, and other elements necessary to establish a global e-Decisional Community. Central to the approach unique Set of Experience knowledge representation and Decisional DNA are at the main focus of our research and this paper.

Keywords: Experience, knowledge representation, Decisional DNA, trust.

1 Introduction and Our Vision

Typically, decisional experiences are not stored, unified, improved, reused, shared, or distributed. This fact motivated the research outlined in this article that aims at capturing, improving and reusing the vast amount of knowledge amassed in past decisional experience.

In nature, deoxyribonucleic acid (DNA) contains “...the genetic instructions used in the development and functioning of all known living organisms. The main role of DNA molecules is the long-term storage of information. DNA is often compared to a set of blueprints and the DNA segments that carry this genetic information are called genes” [1]. The idea behind our research is to develop an artificial system, an architecture that would support discovering, adding, storing, improving and sharing information and knowledge among agents and organisations through experience. We propose a novel Knowledge Representation (KR) approach in which experiential knowledge is represented by Set of Experience (SOE), and is carried into the future by Decisional DNA (DDNA). Using SOE and DDNA, we further establish principles and the concept of global e-Decisional Community and Knowledge Cloud.

Our use of the analogy between the combination of the four nucleotides of DNA and the four components that characterise decision making actions (variables, functions, constraints, and rules) is unique. Some of the principles for such integration and uniqueness from the perspective of capturing experience and history of decision making in an organisation has been mentioned by others (e.g.: *JC Elliott and TJ Elliot (2005) “Decision DNA”, Infinity: PA*) but has never evolved into detailed research

structure and aim as presented here. The DNA metaphor has also been used successfully in terms of formalising organisational knowledge in stipulations of organisation performance (decision rights, information, motivators, and structure). In this sense the proposed Decisional DNA complements and significantly expands these directions of organisational research and adds formalisation of its technological aspects. Using SOEKS and Decisional DNA our aim is to develop a smart experience and knowledge management platform and then expand the developed framework into global e-Decisional collaborative community.

2 Current Research by Others in the Related Area: Review of Relevant Literature

Since the underlying Drucker's [2] proposition knowledge representation systems have been the goal of researchers and plant engineers alike for over 15 years. However, these problems and their solutions do not appear to have progressed too far. The fundamental limitation of current research in this area is that none of the proposed approaches uses experience as *ongoing, real time reference* during the decisional process in a way which happens naturally when humans make decisions if confronted with a new situation. This fact was an instrumental realization at the beginning of our research conceptualization leading to the question "*what's wrong with the current methods and how could they be improved to assure progress?*" Our research approach proposed here answers this question. We challenge the existing techniques with the proposition that all of them lack the same critical element in assuring progress and useful implementations – they don't store and reuse experience in an ongoing, real-time manner. The knowledge management platform we propose is experience based and captures experience on the day-to-day operation; the knowledge representation we introduce combines logic, rules and frames, and it is experience based; our industrial implementations are experience based. This is novelty leading ultimately (through integration of a number of platforms) to fundamentals and raise of global knowledge sharing architecture – the e-Decisional Community and Cloud of Knowledge.

2.1 State of the Art

Our approach is motivated by the following needs and directions specified by current, cutting-edge international research in the area:

- First, acquisition of knowledge through efficient transformation of data and information, and then management of such knowledge, becomes the main challenge of knowledge society [3,4,5,6]. Within this general challenge, there is a very specific need formulated recently by a number of researchers [7,8,9,10] – the need to develop a KM System which acts as a knower, i.e. that "has knowledge, develops it, and applies it" [9]. Trying to answer this need which is yet to be fulfilled, we propose in our research a smart administration system able to store, develop, improve and apply knowledge.

- Second, European and Australian studies reported in [5] have established that the primary research aim of KM should be to use the vast experience that is accumulating each day within organisations, as true knowledge is build up through learning from current and past experiences [3,11,12]. New tools are needed to convert experiences into knowledge that can be improved, accessed and used by decision makers. Our research develops such tools.
- Next, Experience Management (EM) as the basis for knowledge generation and representation is capturing increasingly growing attention of researchers and practitioners, especially in Europe [5,11,12]. However, the existing EM technology acts as a document repository, and is not yet a truly intelligent decision support system. Our approach embarks on such an intelligent decision support system development.
- The existing KM systems are human centred and act as information repositories; they do not act as automatic or semiautomatic decision makers, they do not act as “knowers”. Some of the best examples of the above are ExpertSeeker Web Miner by NASA-GSFC (repository of expertise), ExpertSeeker KSC by NASA-KSC (expertise locator), NaCoDAE by the US Navy (conversational based decisional repository) or Universal Knowledge by KPS (document repository) [13]. We propose to go beyond repository human centre like systems that are currently used by turning the system into a “knower” system.
- Relevant state of the art literature in the area suggests that the question of how to automate experience based knowledge administration using intelligent techniques and software engineering methodologies is still an unsolved research issue [5,10]. Our approach to the solution of this issue is to systematically create, capture, reuse, improve and distribute experience in the work processes of an organization, preventing important decisional steps from being forgotten in the daily operation, and supporting a path towards appropriate automation for recurring tasks or findings.

3 Semantic Technologies and Trustable Knowledge

Semantic technologies constitute one of the most interesting technologies derived from the World Wide Web revolution. It is a field constantly reviewed in different areas of knowledge and its greatest improvements for information and knowledge technologies are still there to be discovered.

According to some members of the scientific community, it is true that the whole concept of the semantic web presented by Tim Berners-Lee in his foundational article [14] is not reached yet; however, the improvements present in today’s Web sites and search engines are not to be underestimated.

Within the myriads of semantic based techniques available, a great attention has been given to ontologies and how their implementation and use enhance real world applications that are not directly related to the Web itself. Ontologies offer great flexibility and capability to model specific domains, and hence, conceptualize the portion of reality to which such domain refers. Nevertheless, it is not enough to have a good

modelled ontology fed with real world instances (individuals) from trustable sources of information; nowadays, it is of the utmost importance to enhance such technologies with decisional capabilities that can offer trustable knowledge in a fast way. On this regard, the introduction of concepts such as the Set of Experience Knowledge Structure (SOEKS or shortly SOE), Decisional DNA [15] and Reflexive Ontologies (RO) [16] lead to alternative technologies that can offer trustable knowledge.

On one hand, the SOE is a knowledge structure that allows the acquisition and storage of formal decision events in a knowledge-explicit form. It comprises variables, functions, constraints and rules associated in a DNA shape allowing the construction of the Decisional DNA of an organization. On the other hand, the RO technique can be used to add self contained queries to an ontology and improves query speed, adds new knowledge about the domain, and allows self containment of the Knowledge Structure in a single file.

Having a powerful knowledge structure such as the Decisional DNA enhanced with the RO technique can be considered as an important advance in the development of knowledge systems. However, the need of trustable knowledge makes necessary to include additional elements in order to achieve what Tim Berners-Lee proposed.

4 Set of Experience Knowledge Structure (SOEKS) and Decisional DNA

Arnold and Bowie [17] argue that “the mind’s mechanism for storing and retrieving knowledge is transparent to us. When we ‘memorize’ an orange, we simply examine it, think about it for a while, and perhaps eat it. Somehow, during this process, all the essential qualities of the orange are stored [experience]. Later, when someone mentions the word ‘orange’, our senses are activated from within [query], and we see, smell, touch, and taste the orange all over again”. The SOEKS has been developed to keep formal decision events in an explicit way [15]. It is a model based upon existing and available knowledge, which must adjust to the decision event it is built from; besides, it can be expressed in OWL (Ontology Web Language) as an ontology in order to make it shareable and transportable [15][18][19]. Four basic components surround decision-making events, and are stored in a combined dynamic structure that comprises the SOE. These four components are variables, functions, constraints, and rules.

Additionally, the SOEKS is organized in a DNA shape. The elements of the structure are connected among themselves imitating part of a long strand of DNA, that is, a gene. Thus, a gene can be assimilated to a SOE, and, in the same way as a gene produces a phenotype, a SOE produces a value of decision in terms of the elements it contains; in other words, the SOEKS, itself, stores an answer to a query presented.

A unique SOE cannot rule a whole system, even in a specific area or category. Therefore, more Sets of Experience should be acquired and constructed. The day-to-day operation provides many decisions, and the result of this is a collection of many

different SOE. A group of SOE of the same category comprises a decisional chromosome, as DNA does with genes. This decisional chromosome stores decisional “strategies” for a category. In this case, each module of chromosomes forms an entire inference tool, and provides a schematic view for knowledge inside an organization. Subsequently, having a diverse group of SOE chromosomes is like having the Decisional DNA of an organization, because what has been collected is a series of inference strategies related to such enterprise.

In conclusion, the SOEKS is a compound of variables, functions, constraints and rules, which are uniquely combined to represent a formal decision event. Multiple SOE can be collected, classified, and organized according to their efficiency, grouping them into decisional chromosomes. Chromosomes are groups of SOE that can accumulate decisional strategies for a specific area of an organization. Finally, sets of chromosomes comprise what is called the Decisional DNA of the organization [15][18].

5 Ontologies, Reflexive Ontologies (RO), and the Semantic Web

Tom Gruber’s [20] accepted definition in the computer science field for an ontology states that it is the explicit specification of a conceptualization; a description of the concepts and relationships in a domain. In the context of Artificial Intelligence (AI), we can describe the ontology of a program by defining a set of representational terms. In such ontology, definitions associate names of entities in the universe of discourse with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms. Computer programs can use ontologies for a variety of purposes including inductive reasoning, classification, and problem solving techniques. In addition, emerging semantic systems use ontologies for a better interaction and understanding between different agent-based systems. Ontologies can be modelled using several languages, being the most widely used RDF and OWL.

Reflexivity addresses the property of an abstract structure of a knowledge base (in this case, an ontology and its instances) to “know about itself”. When an abstract knowledge structure is able to maintain, in a persistent manner, every query performed on it, and store those queries as individuals of a class that extends the original ontology, it is said that such ontology is reflexive. Thus, Toro et al. [16] proposed the following definition for a Reflexive Ontology: “A Reflexive Ontology is a description of the concepts and the relations of such concepts in a specific domain, enhanced by an explicit self contained set of queries over the instances”. Therefore, any RO is an abstract knowledge structure with a set of structured contents and relationships, and all the mathematical concepts of a set can be applied to it as a way of formalization and handling.

The advantage of implementing RO relies on the following main aspects: Speed on the query process, incremental nature, and self containment of the knowledge structure in a single file.

The World Wide Web (WWW) was created less than two decades ago, and in such short time, it has developed with an astonishing speed. It allows us to communicate

and exchange information and knowledge all over the world in a way that was unthinkable some years before its creation. Furthermore, in 2001, Tim Berners-Lee [14] proposed a concept called the Semantic Web and offered a future vision of the WWW where information is understandable not only by humans but also by machines.

In the Semantic Web proposal, applications make use of knowledge in order to gain automation of tasks that are currently performed with heavy user interaction. Semantic Web systems will require new components of knowledge representation and semantic technologies immerse in web environments with intelligent capabilities. It will presuppose the existence of a common vocabulary shared by all the agents in the semantic system. Semantic Web applications may need semantic models that will enable it to draw conclusions and/or take decisions. Ontologies can be considered as one of these models. New technologies are bringing the Semantic Web vision into fruition, opening doors to new web-based applications, ranging from semantic search engines to intelligent agents.

6 Security and Trust

In terms of information and computer fields, it is common to use information security, computer security or information assurance. Information security means protecting information and its systems from unauthorized access, use, disclosure, disruption, modification, or destruction [21]. These fields are cross related and share common goals of protecting confidentiality, integrity and availability of information.

All sort of organizations, both public and private collect a great deal of confidential information about their employees, customers, products, research, and financial status. Most of this information is gathered, processed, stored and transmitted across networks of computers. Protecting confidential information is nowadays a business obligation.

The field of information has grown and evolved significantly in recent years to the point of being transformed into knowledge. Moreover, it is frequently to read the term trust as an upper level of security. Thus, in the knowledge society, the requirement of not just knowledge security, but knowledge trust, becomes a main issue.

In this paper, we are not relaying nor trying to have a single definition of trust; nevertheless, we are going to mention some models that scientists have introduced in order to offer security and trust. As commonly done, we perceive trust as an additional element or an output of security which is mathematically understood as a probability of expected positive behaviour.

Numerous approaches to handle trust relations have been presented. Dewan and Dasgupta [22] propose a strategy based on reputations. The reputation ratio is related to the positive or negative behaviour of an entity. Marsh [23] model of trust was adapted by Pirzada and McDonald [24]. Their trust value calculation is based upon a weight value of the transaction. Such weight value of a transaction is defined according to the benefit received for the entity. Additionally, Beth et al. [25] introduces the computation of trust based on recommendations. Recommendation trust is established

upon a transitive trust relation to an unknown entity that is given for a third trusted party. Also based on recommendations, Josang [26] presents a trust value with the help of subjective probability. For him, trust is represented as an opinion which is calculated out of the positive and negative experiences concerning the target of the opinion.

When referring to decisions, trustable knowledge implies the establishment of guidelines for making reliable decisions, that is, trust-related decisions, such as who produces the knowledge, how trustworthy to be, what knowledge to believe, and how truthful to be when recommending knowledge. This is what we refer to as Decisional Trust.

7 Decisional Trust

Decisional Trust relies on three elements: the Decisional DNA, Reflexive Ontologies and Trust Technologies.

The Decisional DNA offers adaptability on gathering, storing and managing decisional knowledge. It is a strong knowledge structure able to support diverse decisional elements at all levels. User modelling, task, knowledge and experience are possible scenarios for the exploitation of the Decisional DNA. Moreover, Decisional DNA has proven to be a useful mathematical and logical inference tool on decision making and knowledge management.

Furthermore, generally, any knowledge is subject to be modelled as an ontology. From our research experience, a good starting point is to have a well defined schema with some general elements in the area of domain that is being described, in our case, the Decisional DNA. One of the advantages of a conceptual knowledge model expressed as an ontology is the capacity of inferring semantically new derived queries. These queries relate concepts that are not explicitly specified by the user; nevertheless the concepts are relevant to the query. Modern inference engines and reasoners like Pellet and Racer deliver a highly specialized, yet efficient way to perform such queries via a JAVA compliant API. In the literature, data handling by ontology-based technology is reported by researchers in different fields [19][27][16]. Then, if a knowledge structure such as the Decisional DNA is enhanced with the capabilities of ontology based technology, its performance is increased in terms of two characteristics: *complementary inference capabilities* added by the inference engines of ontology technologies; and *share abilities* given by the semantic annotation meta-languages in which ontologies are transmitted. Adding heavier semantics, logic, and expressiveness to the Decisional DNA resulted in an OWL decisional Ontology. However, we propose to broaden even more the Decisional DNA ontology with the capabilities of a Reflexive Ontology profiting in performance for its additional properties.

Finally, all this knowledge is boosted with trust technologies transforming such knowledge into Decisional Trust within the Semantic Web. For more information see [28].

8 Implementing Decisional Trust

Establishing decisional trust strategies is a complex problem. Experiences provide any agent with a trust-strategy with trustworthiness feedback that is certain and as such, we consider it as the base of our Decisional Trust. Learning to trust based on numerous repeated learning processes is advantageous. Our focus lies on the question how far a recommended trust-experience or trust-knowledge is suitable to be the base of a trust-decision (i.e. new experience or knowledge). Thus, we propose the idea that experience is essential for a direct-trust relation between a previous experience and a future decision. Our solution tries to condense the chains of recommendation to only one value (i.e. the trust value), but maintains the knowledge untouched. We deal with trust in terms of the user who originated the experience and modify the trust value according to feedback of recommended experience that affects the knowledge. Now, when an agent is taking a decision, the algorithm prefers experiences with a higher trust value.

In regards to the Decisional DNA, the SOEKS within the RO has been enhanced with a trust value. Such trust value is in charge of collecting the chain of decisional experience in terms of trustiness. No recommended experiences but only new individual experiences may lead to new trust.

Toro et al. [29] presented the UDKE platform which has been used as a case study for the use of the Decisional Trust. In this case, a maintenance officer during a routine patrol takes a palmtop device with a camera attached to it. For every object to be maintained, a related Virtual Reality marker can be found (following the sensor concept in ambient intelligence). When the system finds a marker, the matching element to be maintained is identified, and a set of information is extracted from the Decisional DNA with Reflexive Ontologies. The output video stream of the camera is mixed with 3D objects and other relevant information that is displayed to the user in the screen (Figure 1). The exploitation of the experience and its value of trust arise when, for example, the maintenance worker having received the information about changing the extinguisher (i.e. recharging) in December, s/he recommends to do it now (September). The maintenance officer decides to recommend the change due to his/her acquired experience, and it is now being transferred to the system; therefore, according to the user and the previous experiences, the value of trust is transformed.



Fig. 1. AR Enhanced user view example with trust value

9 Conclusion

In this paper, the concept of Decisional Trust is presented. A schema based upon three main technologies is explained as the means to achieve Decisional Trust: the Decisional DNA, Reflexive Ontologies and security technologies. The Decisional Trust system is implemented for the exploitation of embedded knowledge in the domain of industrial maintenance in a mobile context, using AR techniques.

This Semantic Web technology could support decisional knowledge and deliver knowledge and trust within the agents that share the technology advancing the current trends of knowledge engineering.

Further research would expand the operations of the proposed Knowledge Engineering tools towards creating e-Decisional Community and Cloud of Knowledge. At this next stage, we plan to develop an architecture supporting a number of collaborating platforms as a grid of knowledge network which facilitates companies to share, improve and create knowledge, resulting in greater efficiency, effectiveness, and an increase of collective intelligence of the whole community. Ideas related to collective intelligence, social networks and conflict resolution would be investigated as part of this future research direction [30,31,32]. The Cloud of Knowledge conceptual vision was very recently published by our Knowledge Engineering Research Team (KERT) in [33].

References

1. Sinden, R.R.: DNA Structure and Function. Academic Press, San Diego (1994)
2. Drucker, P.: The Post-Capitalist Executive: Managing in a Time of Great Change. Penguin, New York (1995)
3. Awad, E.M., Ghaziri, H.M.: Knowledge Management. Prentice Hall, New Jersey (2004)
4. Sun, Z., Finnie, G.: Experience Management in Knowledge Management. In: Khosla, R., Howlett, R., Jain, L. (eds.) KES 2005. LNCS (LNAD), vol. 3681, pp. 979–986. Springer, Heidelberg (2005)
5. Boahene, M., Ditsa, G.: Conceptual Confusions in Knowledge Management and Knowledge Management Systems: Clarifications for Better KMS Development. In: Knowledge Management: Current Issues and Challenges, IRM Press, London (2003)
6. Hakanson, Hartung: Using Reengineering for Knowledge Based Systems. Cybernetics and Systems 38, 799–824 (2007)
7. Hasan, H., Handzic, M.: Australian Studies in Knowledge Management. University of Wollongong Press, Wollongong (2003)
8. Mitchell, H.: Technology and Knowledge Management: Is Technology Just an Enabler or Does it also Add Value? In: Knowledge Management: Current Issues and Challenges. IRM Press, London (2004)
9. Yang, Reidsema: Information handling in a knowledge based intelligent design system. Cybernetics and Systems 30, 549–574 (2008)
10. Althoff, K.-D., Decker, B., Hartkopf, S., Jedlitschka, A., Nick, M., Rech, J.: Experience Management: The Fraunhofer IESE Experience Factory. In: Industrial Conference Data Mining, Berlin (2001)
11. Bergmann, R.: Experience Management. Springer, New York (2004)

12. Universal Knowledge (2010),
<http://www.kpsol.com/products/universal/index.html/>
(accessed August 01, 2010)
13. Weidenhausen, J., Knoepfle, C., Stricker, D.: Lessons learned on the way to industrial augmented Reality applications, a retrospective on ARVIKA. *Computers and Graphics* 27(6), 887–891 (2004)
14. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American* 5(284), 28–31 (2001)
15. Sanin, C., Szczerbicki, E.: Set of Experience: A Knowledge Structure for Formal Decision Events. *Foundations of Control and Management Sciences Journal* 3, 95–113 (2005)
16. Toro, C., Sanín, C., Szczerbicki, E., Posada, J.: Reflexive Ontologies: Enhancing Ontologies with Self- Contained Queries. *International Journal of Cybernetics and Systems* 39(2), 171–189 (2007)
17. Arnold, W., Bowie, J.: *Artificial Intelligence: A Personal Commonsense Journey*. Prentice Hall, New Jersey (1985)
18. Sanin, C., Szczerbicki, E.: Using XML for Implementing Set of Experience Knowledge Structure. In: Khosla, R., Howlett, R., Jain, L. (eds.) *KES 2005. LNCS (LNAI)*, vol. 3681, pp. 946–952. Springer, Heidelberg (2005)
19. Sanin, C., Toro, C., Szczerbicki, E.: An OWL ontology of set of experience knowledge structure. *Journal of Universal Computer Science* 13(2), 209–223 (2007)
20. Gruber, T.R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies* 43(5-6), 907–928 (1995)
21. Cornell University law School, U.S. Code Collection,
http://www.law.cornell.edu/uscode/html/uscode44/uscode44/uscode44_0003542--000-.html (retrieved on February 2008)
22. Dewan, P., Dasgupta, P.: Trusting Routers and Relays in Ad hoc Networks. In: *First International Workshop on Wireless Security and Privacy (WiSr 2003) in Conjunction with IEEE 2003 International Conference on Parallel Processing Workshops (ICPP)*, Kaho-siung, Taiwan, pp. 351–358 (October 2003)
23. Marsh, S.: *Formalising Trust as a Computational Concept*. PhD Thesis, University of Stirling, UK (1994)
24. Pirzada, A., McDonald, C.: Establishing Trust in Pure Ad-hoc Networks. In: *Proceedings of the 27th Conference on Australasian Computer Science (ACSC 2004)*, Dunedin, New Zealand, vol. 26, pp. 47–54 (2004)
25. Beth, T., Borcherding, M., Klein, B.: Valuation of Trust in Open Networks. In: Gollmann, D. (ed.) *ESORICS 1994. LNCS*, vol. 875, pp. 3–18. Springer, Heidelberg (1994)
26. Josang, A.: A Subjective Metric of Authentication. In: Quisquater, J.-J., Deswarte, Y., Meadows, C., Gollmann, D. (eds.) *ESORICS 1998. LNCS*, vol. 1485, pp. 329–344. Springer, Heidelberg (1998)
27. Sevilimis, N., Stork, A., Smithers, T., Posada, J., Pianciamore, M., Castro, R., Jimenez, I., Marcos, G., Mauri, M., Selvini, P., Thelen, B., Zecchino, V.: Knowledge Sharing by Information Retrieval in the Semantic Web. In: Gómez-Pérez, A., Euzenat, J. (eds.) *ESWC 2005. LNCS(LNAI)*, vol. 3532, pp. 471–485. Springer, Heidelberg (2005)
28. Sanin, C., Szczerbicki, E., Toro, C.: Towards a technology of trust: Reflexive Ontologies and Decisional DNA. In: *1st International IEEE Conference on Information Technology*, Gdansk, Poland, pp. 31–34 (May 2008)

29. Toro, C., Sanín, C., Vaquero, J., Posada, J., Szczerbicki, E.: Knowledge based industrial maintenance using portable devices and augmented reality. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part I. LNCS (LNAI), vol. 4692, pp. 295–302. Springer, Heidelberg (2007)
30. Hwang, D., Nguyen, N.T., et al.: A Semantic Wiki Framework for Reconciling Conflict Collaborations Based on Selecting Consensus Choice. *Journal of Universal Computer Science* 16(7), 1024–1035 (2010)
31. Duong, T.H., Nguyen, N.T., Jo, G.S.: Constructing and Mining: A Semantic-Based Academic Social Network. *Journal of Intelligent & Fuzzy Systems* 21(3), 197–207 (2010)
32. Nguyen, N.T.: Processing Inconsistency of Knowledge in Determining Knowledge of a Collective. *Cybernetics and Systems* 40(8), 670–688 (2009)
33. Mancilla, L., Sanin, C., Szczerbicki, E.: Using human behaviour to develop knowledge-based virtual organisations. *Cybernetics and Systems: An International Journal* 41(8), 577–591 (2010)

Overview of Algorithms for Swarm Intelligence

Shu-Chuan Chu¹, Hsiang-Cheh Huang², John F. Roddick¹, and Jeng-Shyang Pan³

¹ School of Computer Science, Engineering and Mathematics,
Flinders University of South Australia, Australia

² National University of Kaohsiung, 700 University Road, Kaohsiung 811, Taiwan, R.O.C.

³ National Kaohsiung University of Applied Sciences, 415 Chien-Kung Road,
Kaohsiung 807, Taiwan, R.O.C.

Abstract. Swarm intelligence (SI) is based on collective behavior of self-organized systems. Typical swarm intelligence schemes include Particle Swarm Optimization (PSO), Ant Colony System (ACS), Stochastic Diffusion Search (SDS), Bacteria Foraging (BF), the Artificial Bee Colony (ABC), and so on. Besides the applications to conventional optimization problems, SI can be used in controlling robots and unmanned vehicles, predicting social behaviors, enhancing the telecommunication and computer networks, etc. Indeed, the use of swarm optimization can be applied to a variety of fields in engineering and social sciences. In this paper, we review some popular algorithms in the field of swarm intelligence for problems of optimization. The overview and experiments of PSO, ACS, and ABC are given. Enhanced versions of these are also introduced. In addition, some comparisons are made between these algorithms.

Keywords. Swarm intelligence (SI), Particle Swarm Optimization (PSO), Ant Colony System (ACS), Artificial Bee Colony (ABC).

1 Introduction

People learn a lot from Mother Nature. Applying the analogy to biological systems with lots of individuals, or swarms, we are able to handle the challenges in the algorithm and application with optimization techniques. In this paper, we focus on the overview of several popular swarm intelligence algorithms, pointing out their concepts, and proposing some enhancements of the algorithms with the results of our research group.

Swarm intelligence, according to [1], is the emergent collective intelligence of groups of simple agents. With swarm intelligence, the developed algorithms need to be flexible to internal and external changes, to be robust when some individuals fail, to be decentralized and self-organized [2]. In the rest of the paper, we will address several popular algorithms based on these concepts, including Particle Swarm Optimization (PSO), Ant Colony System (ACS), and Artificial Bee Colony (ABC) algorithms in Sec. 2, and we present the improvements of these algorithms based on our existing works in Sec. 3. Selected simulation results and comparisons are also provided in Sec. 4. Finally, we conclude this paper in Sec. 5.

2 Swarm Intelligence Algorithms

In this section, we introduce the concept and implementation of several popular algorithm for swarm intelligence optimization, including particle swarm optimization (PSO), ant colony system (ACS), and Artificial Bees Colony (ABC) algorithms.

2.1 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) was first introduced by Kennedy and Eberhart [3,4]. It is a relatively new stochastic optimization technique that can simulate the swarm behavior of birds flocking. In PSO, an individual in the swarm, called a particle, represents a potential solution. Each particle has a fitness value and a velocity, and it learns the experiences of the swarm to search for the global optima [5]. Traditional PSO can be depicted in Fig. 1. They include (1) particle initialization, (2) velocity updating, (3) particle position updating, (4) memory updating, and (5) termination checking. These steps are described as follows.

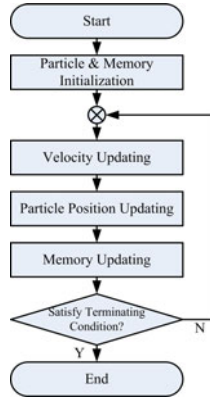


Fig. 1. Procedures for particle swarm optimization

- (1) *Initialization.* We first decide how many particles used to solve the problem. Every particle has its own position, velocity and best solution. If we use M particles, their best solutions, and their velocities can be represented as:

$$\mathbf{X} = \{x_0, x_1, \dots, x_{M-1}\}, \quad (1)$$

$$\mathbf{B} = \{b_0, b_1, \dots, b_{M-1}\}, \quad (2)$$

$$\mathbf{V} = \{v_0, v_1, \dots, v_{M-1}\}. \quad (3)$$

- (2) *Velocity updating.* This step is shown in Eq. (4), where c_1 and c_2 are constants, r_1 and r_2 are random variables in the range from 0 to 1, $b_i(t)$ is the best solution of the i -th particle for the iteration number up to the t -th iteration and the $G(t)$ is the best solution of all particles:

$$v_i(t+1) = v_i(t) + c_1 \cdot r_1 \cdot (b_i(t) - x_i(t)) + c_2 \cdot r_2 \cdot (G(t) - x_i(t)). \quad (4)$$

To prevent the velocity from becoming too large, we set a maximum value to limit the range of velocity as $-V_{MAX} \leq V \leq V_{MAX}$.

- (3) *Position updating*, which is processed by Eq. (5):

$$x_i(t+1) = x_i(t) + v_i(t), \quad i = 0, 1, \dots, M-1. \quad (5)$$

- (4) *Memory updating*. If we find a better solution than $G(t)$ in $G(t+1)$, $G(t)$ will be replaced by $G(t+1)$. Otherwise, there will be no change for $G(t)$.

- (5) These recursive steps continue unless we reach the termination condition.

With the descriptions above, we can observe that the solution in PSO can be influenced by both the global and the local characteristics in the training procedure.

2.2 Any Colony Systems

Inspired by the food-seeking behavior of real ants, the ant system [6,7] is a cooperative population-based search algorithm. As each ant constructs a route from nest to food by stochastically following the quantities of pheromone level, the intensity of laying pheromone would bias the path-choosing, decision-making of subsequent ants.

The operation of ant system can be illustrated by the classical traveling salesman problem (TSP). The TSP seeks for a round route covering all cities with minimal total distance. Suppose there are n cities and m ants. The entire algorithm starts with initial pheromone intensity set to s_0 on all edges. In every subsequent ant system cycle, or the episode, each ant begins its tour from a randomly selected starting city and is required to visit every city once and only once. The experience gained in this phase is then used to update the pheromone intensity on all edges.

The algorithm of the ant system for the TSP is depicted as follows [7,8]:

- (1) Randomly select the initial city for each ant. The initial pheromone level between any two cities is set to be a small positive constant. Set the cycle counter to be 0.
- (2) Calculate the transition probability from city r to city s for the k -th ant as

$$P_k(r, s) = \begin{cases} \frac{[\tau(r, s)] \cdot [\eta(r, s)]^\beta}{\sum_{u \in J_k(r)} [\tau(r, u)] \cdot [\eta(r, u)]^\beta}, & \text{if } s \in J_k(r), \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where r is the current city, s is the next city, $\tau(r, s)$ is the pheromone level between cities r and s , $\eta(r, s) = \delta(r, s)^{-1}$ is the inverse of the distance $\delta(r, s)$ between cities r and s , $J_k(r)$ is the set of cities that remain to be visited by the k -th ant positioned on city r , and β is a parameter determining the relative importance of pheromone level versus distance. Select the next visited city s for the k -th ant with the probability $P_k(r, s)$. Repeat Step (2) for each ant until the ants have toured all cities.

- (3) Update the pheromone level between cities as

$$\tau(r, s) \leftarrow (1 - \alpha) \cdot \tau(r, s) + \sum_{k=1}^m \Delta \tau_k(r, s), \quad (7)$$

$$\Delta \tau_k(r, s) = \begin{cases} \frac{1}{L_k}, & \text{if } (r, s) \in \text{route done by ant } k, \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

where $\Delta \tau_k(r, s)$ is the pheromone level laid down between cities r and s by the k -th ant, L_k is the length of the route visited by the k -th ant, m is the number of ants and $0 < \alpha < 1$ is a pheromone decay parameter.

- (4) Increment cycle counter. Move ants to originally selected cities and continue Steps (2)–(4) until the behavior stagnates or the maximum number of cycles has reached. A stagnation is indicated when all ants take the same route.

From Eq. (6) it is clear ant system (AS) needs a high level of computation to find the next visited city for each ant. In order to improve the search efficiency and lower computational complexity, the ant colony system (ACS) was proposed [8,9]. ACS is based on AS but updates the pheromone level before moving to the next city (local updating rule) and updating the pheromone level for the shortest route only after completing the route for each ant (global updating rule) as

$$\tau(r, s) \leftarrow (1 - \alpha) \cdot \tau(r, s) + \alpha \cdot \Delta \tau(r, s), \quad (9)$$

$$\Delta \tau(r, s) = \begin{cases} \frac{1}{L_{gb}}, & \text{if } (r, s) \in \text{global best route,} \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

where L_{gb} is the length of the shortest route and α is a pheromone decay parameter.

2.3 Artificial Bee Colony (ABC) Optimization

Karaboga proposed Artificial Bee Colony (ABC) [10] in 2005 based on inspecting the behaviors of real bees on finding nectar and sharing the information of food sources to the bees in the nest. Three kinds of bees are defined as the artificial agents. Every kind of bee plays different and important roles in the optimization process. The artificial agents are called the employed bee, the onlooker, and the scout with distinct responsibilities. The employed bee stays on a food source, which represents a spot in the solution space, and provides the coordinate for the onlookers in the hive for reference. The onlooker bee receives the locations of food sources and selects one of the food sources to gather the nectar. The scout bee moves in the solution space to discover new food sources. The process of the ABC optimization is described as follows:

- (1) *Initialization*. Spray n_e percentage of the populations into the solution space randomly, and then calculate their fitness values, called the nectar amounts, where n_e represents the ratio of employed bees to the total population. Once these populations are positioned into the solution space, they are called the

employed bees. Evaluate the fitness of the employed bees and take the fitness to be their amount of nectar.

- (2) Move the onlookers. We calculate the probability of selecting a food source by Eq. (9) first, where θ_i denotes the position of the i -th employed bee, $F(\bullet)$ is the fitness function, S represents the number of employed bees, and P_i is the probability of selecting the i -th employed bee. Then we select a food source to move to by roulette wheel selection for every onlooker bee and determine the nectar amounts. The onlookers are moved by Eq. (10), where x_i denotes the position of the i -th onlooker bee, t denotes the iteration number, θ_k is the randomly chosen employed bee, j represents the dimension of the solution, and $\phi(\bullet)$ produces a series of random variable in the range $[-1, 1]$.

$$P_i = \frac{F(\theta_i)}{\sum_{k=1}^S F(\theta_k)}, \quad (11)$$

$$x_{ij}(t+1) = \theta_{ij}(t) + \phi(\theta_{ij}(t) - \theta_{kj}(t)). \quad (12)$$

- (3) Update the best food source found so far. We record the best fitness value and the position, which are found by the bees.
- (4) Move the scouts. If the fitness values of the employed bees are not improved by a consecutive number of iterations, called "Limit," those food sources are abandoned, and these employed bees become the scouts. The scouts are moved by Eq. (11), where r is a random number and $r \in [0, 1]$.

$$\theta_{ij} = \theta_{j\min} + r \cdot (\theta_{j\max} - \theta_{j\min}) \quad (13)$$

- (5) Termination checking. If the amount of the iterations satisfies the termination condition, we terminate the program and output the results; otherwise, go back to Step (2).

2.4 Relating Algorithms with SI

Due to the limited space in this paper, readers who are interested in relating algorithms are suggested to refer to the followings: Bacteria Foraging (BF) [11], Cat Swarm Optimization (CSO) [14,13], and Stochastic Diffusion Search (SDS) [14].

3 The Enhanced Swarm Intelligence Algorithms

3.1 Parallel PSO

A parallel computer consists of a large number of processing elements which can be dedicated to solving a single problem at a time. Pipeline processing and data parallelism are two popular parallel processing methods. The function of the pipeline processing is to separate the problem into a cascade of tasks where each task is executed by an individual processor, while data parallelism involves distributing the

data to be processed amongst all processors which then executes the same procedure on each subset of the data. And this is the motivation for the parallel PSO.

Computation time of PSO in Sec. 2.1 can be reduced with the parallel structure. Parallel processing aims at producing the same results achievable using multiple processors with the goal of reducing the run time. We also have the same steps in describing parallel PSO (PPSO) in Sec. 2.1. But in Step (1), users must define how many groups need be in the process because it can be designed to be 2^n sets. To decide the number of groups, it should be carefully refer to the size of the problem to solve. In general, we use four groups to evolve, but we use eight groups only if the problem is quite large. After several iterations, each group exchanges information in accordance with an explicit rule chosen by users.

The parallel particle swarm optimization (PPSO) method [15] gives every group of particles the chance to have the global best and the local best solutions of other groups. It increases the chance of particles to find a better solution and to leap the local optimal solutions.

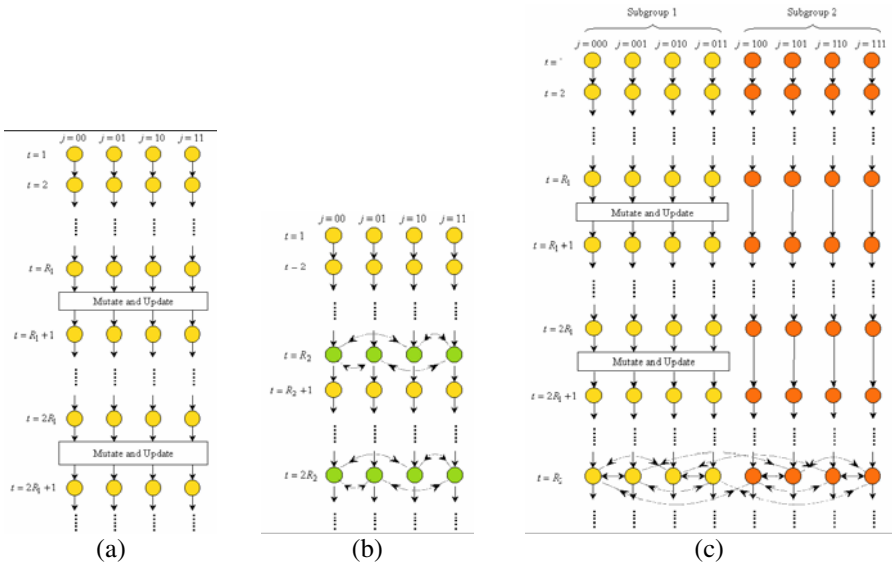


Fig. 2. Communication strategies for parallel PSO. (a) Strategy #1, with loose correlation. (b) Strategy #2, with strong correlation. (c) Strategy #3, with general purpose usage.

Three communication strategies for PPSO are presented. The first one is to deal with the loosely correlated parameters in functions. When all the particles evolved a period of time (Here we described as R_1 iterations), communication strategy 1 would exchange the local best solution from one group to others. The way of communication strategy 1 is shown in Fig. 2(a).

The second communication strategy is the self-adjustment in each group due to the strong correlations of parameters. In each group, the best particle would migrate to its neighborhood to supplant some particles by replacing a deteriorated solution with the

group's best one. For the ease of implementation, the number of groups for the parallel structure is set to a power of two. Thus, neighborhoods are defined as one bit difference in the binary format. The second communication strategy would be applied every R_2 iterations for replacing the poorly performing particles. The way of communication strategy 2 is shown in Fig. 2(b).

The third communication strategy of PPSO is the combination of communication strategies 1 and 2. The particles must be separate into at least four groups. In communication strategy 3, all particles are separated into two subgroups. Under the subgroups, communication strategy 1 and 2 are imitated. Thus, communication strategy 3 is a general communication strategy for PPSO. The process is shown in Fig. 2(c).

Based on the observation, if the parameters are loosely correlated or independent, communication strategy 1 will obtain good results quite quickly. On the contrary, communication strategy 2 obtains higher performance if the parameters are tightly correlated. However, these communication strategies may perform poorly if they have been applied in the wrong situation. Consequently, when the correlation of parameters is unknown, communication strategy 3 is the better choice to apply.

3.2 Parallel ACS

Similar to PPSO, we apply the idea of data parallelism to ant colony system (ACS) in Sec. 2.2 to reduce running time and obtain a better solution. The parallel ant colony system (PACS) based on traveling salesman problem is described as follows:

- (1) *Initialization.* Generate N_j artificial ants for the j -th group, $j = 0, 1, \dots, G - 1$, and G is the number of groups. Randomly select an initial city for each ant. The initial pheromone level between any two cities is set to be a small positive constant τ_0 . Set the cycle counter to be 0.
- (2) *Movement.* Calculate the next visited city s for the i -th ant in the j -th group according to

$$s = \arg \max_{u \in J_{i,j}(r)} [\tau_j(r, u)] \cdot [\eta(r, u)]^\beta \quad \text{if } q \leq q_0 \text{ (exploitation)} \quad (14)$$

and visit city s with $P_{i,j}(r, s)$ if $q > q_0$ (biased exploration);

$$P_{i,j}(r, s) = \begin{cases} \frac{[\tau_j(r, s)] \cdot [\eta(r, s)]^\beta}{\sum_{u \in J_k(r)} [\tau_j(r, u)] \cdot [\eta(r, u)]^\beta}, & \text{if } s \in J_{i,j}(r), \\ 0, & \text{otherwise,} \end{cases} \quad (15)$$

where $P_{i,j}(r, s)$ is the transition probability from city r to city s for the i -th ant in the j -th group. $\tau_j(r, s)$ is the pheromone level between city r to city s in the j -th group. $\eta(r, s) = \delta(r, s)^{-1}$ the inverse of the distance $\delta(r, s)$ between city r and city s . $J_{i,j}(r)$ is the set of cities that remain unvisited by the i -th ant in the j -th group and β is a parameter which determines the relative importance of pheromone level versus distance. q is a random number between 0 and 1 and q_0 is a constant between 0 and 1.

- (3) *Local pheromone level updating rule.* Update the pheromone level between cities for each group as

$$\tau_j(r, s) \leftarrow (1 - \alpha) \cdot \tau_j(r, s) + \rho \cdot \Delta\tau(r, s), \quad (16)$$

$$\Delta\tau(r, s) = \tau_0 = \frac{1}{n \cdot L_m}, \quad (17)$$

where $\tau_j(r, s)$ is the pheromone level between cities r and s for the ants in the j -th group, L_m is an approximate distance of the route between all cities using the nearest neighbor heuristic, n is the number of cities and $0 < \rho < 1$ is a pheromone decay parameter. Continue Steps 2 and 3 until each ant in each group completes the route.

- (4) *Evaluation.* Calculate the total length of the route for each ant in each group.
 (5) *Global pheromone level updating rule.* Update the pheromone level between cities for each group as

$$\tau_j(r, s) \leftarrow (1 - \alpha) \cdot \tau_j(r, s) + \rho \cdot \Delta\tau_j(r, s), \quad (18)$$

$$\Delta\tau_j(r, s) = \begin{cases} \frac{1}{L_j}, & \text{if } (r, s) \in \text{best route of } j\text{-th group,} \\ 0, & \text{otherwise,} \end{cases} \quad (19)$$

where L_j is the shortest length for the ants in the j -th group and α is a pheromone decay parameter.

- (6) *Updating from communication.* Seven strategies are proposed as follows:

- I. As shown in Fig. 3(a), update the pheromone level between cities for each group for every R_1 cycles as

$$\tau_j(r, s) \leftarrow \tau_j(r, s) + \lambda \cdot \Delta\tau_{\text{best}}(r, s), \quad (20)$$

$$\Delta\tau_{\text{best}}(r, s) = \begin{cases} \frac{1}{L_{\text{gb}}}, & \text{if } (r, s) \in \text{best route of all groups,} \\ 0, & \text{otherwise,} \end{cases} \quad (21)$$

where k is a pheromone decay parameter and L_{gb} is the length of the best route of all groups, i.e. $L_{\text{gb}} \leq L_j$, $j = 0, 1, \dots, G - 1$.

- II. As shown in Fig. 3(b), update the pheromone level between cities for each group for every R_2 cycles as

$$\tau_j(r, s) \leftarrow \tau_j(r, s) + \lambda \cdot \Delta\tau_{\text{ng}}(r, s), \quad (22)$$

$$\Delta\tau_{\text{ng}}(r, s) = \begin{cases} \frac{1}{L_{\text{ng}}}, & \text{if } (r, s) \in \text{best route of neighbor groups,} \\ 0, & \text{otherwise,} \end{cases} \quad (23)$$

where neighbor is defined as being the group whose binary representation of the group number j differs by the least significant bit. λ is a pheromone decay parameter and L_{ng} is the length of the shortest route

in the neighbor group.

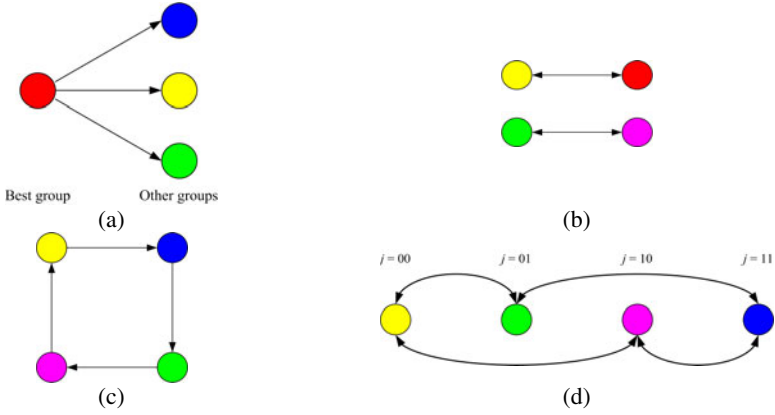


Fig. 3. Updating schemes for communication. (a) Update the pheromone level between cities for each group for every R_1 cycles. (b) Update the pheromone level between cities for each group for every R_2 cycles. (c) Update the pheromone between cities for each group for every R_3 cycles. (d) Update the pheromone between cities for each group for every R_4 cycles.

III. As shown in Fig. 3(c), update the pheromone between cities for each group for every R_3 cycles as

$$\tau_j(r, s) \leftarrow \tau_j(r, s) + \lambda \cdot \Delta\tau_{\text{ng}}(r, s), \quad (24)$$

$$\Delta\tau_{\text{ng}}(r, s) = \begin{cases} \frac{1}{L_{\text{ng}}}, & \text{if } (r, s) \in \text{best route of neighbor groups,} \\ 0, & \text{otherwise,} \end{cases} \quad (25)$$

where neighbor is defined as being the group arranged as the ring structure. λ is a pheromone decay parameter and L_{ng} is the length of the shortest route in the neighbor group.

IV. As shown in Fig. 3(d), update the pheromone between cities for each group for every R_4 cycles as

$$\tau_j(r, s) \leftarrow \tau_j(r, s) + \lambda \cdot \Delta\tau_{\text{ng}}(r, s), \quad (26)$$

$$\Delta\tau_{\text{ng}}(r, s) = \begin{cases} \frac{1}{L_{\text{ng}}}, & \text{if } (r, s) \in \text{best route of neighbor groups,} \\ 0, & \text{otherwise,} \end{cases} \quad (27)$$

where neighbor is defined as being those groups where the binary representation of the group number j differs by one bit. λ is a pheromone decay parameter and L_{ng} is the length of the shortest route in the neighbor group.

- V. Update the pheromone between cities for each group using both Strategies 1 and 2.
 - VI. Update the pheromone between cities for each group using both Strategies 1 and 3.
 - VII. Update the pheromone between cities for each group using both Strategies 1 and 4.
- (7) Termination. Increment the cycle counter. Move the ants to the originally selected cities and continue Steps 2–6 until the stagnation or a present maximum number of cycles has reached, where a stagnation indicated by all ants taking the same route.

With the parallel formulation for the ant colony system, seven communication strategies between groups which can be employed to update the pheromone levels are presented. Details of the parallel ant colony system (PACS) can be observed in [9] with better results for the traveling salesman problem.

3.3 ABC with Communication Strategy

Regarding to Sec. 2.3, to use forward-communication strategy in ABC optimization, the artificial agents in ABC are split into G subpopulations. Each subpopulation evolves by ABC optimization independently, or the subpopulation has its own near best solution, and the artificial agents do not know there are other subpopulations in the solution space. The total iteration contains R times of forward-communications, where $R = \{R_1, 2R_1, 3R_1, \dots\}$. The diagram of the parallelized structure of ABC optimization is given in Fig. 4; the diagram of $k = 1$ forward-communication strategy is given in Fig. 5.

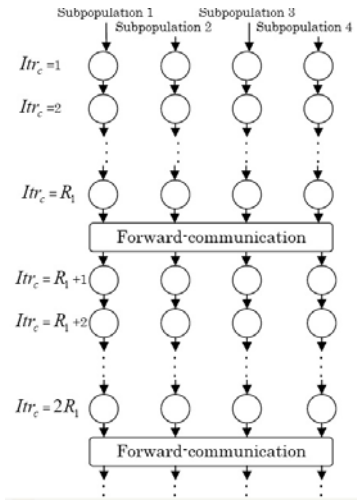


Fig. 4. The diagram of ABC optimization

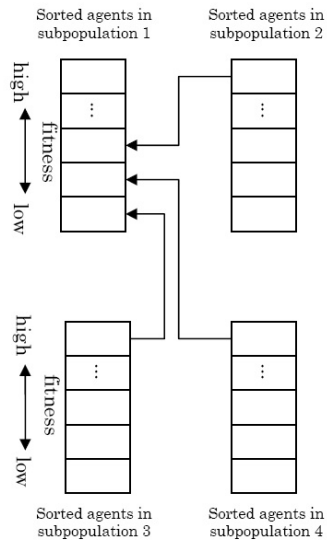


Fig. 5. The diagram of the forward-communication.

When the forward-communication process is executed, the predefined first subpopulation receives $(G-1) \times k$ near best solutions from other subpopulations, where k is the number of agents picked to replace the worst agents in the first subpopulation. The first subpopulation adopts the received near best solutions and wipes out the same number of the worst agents in its population. The subpopulations pop k near best solutions ahead to the first subpopulation in the forward-communication process, and then the subpopulations return to the status with no interaction again. The process of ABC with forward-communication strategy, called ABC-FC, is described as follows:

- (1) *Initialization.* Generate the artificial agents and divide them into G subpopulations. Each subpopulation is initialized by ABC independently. Defined the iteration set R for executing the forward-communication.
- (2) *Evolution of the subpopulations.* Evolve the subpopulations independently by ABC optimization.
- (3) *Forward-communication.* Let Itr_c denotes the current iteration. If $Itr_c \cap R \neq \emptyset$, replace the worst $(G-1) \times k$ agents in the first subpopulation by the near best solutions collected from other subpopulations. Otherwise, skip this step.
- (4) *Termination checking.* If the termination condition is satisfied, output the result and terminate the program. Otherwise go back to Step (2).

With the proposed ABC with forward-communication strategy, it splits the artificial agents into independent subpopulations and provides the forward-communication to the predefined first subpopulation. Better results can be observed in [16].

4 Some Comparisons

Three well-known test functions, namely, Schwefel's function, Rastrign function, and Griewank function, are employed to test the accuracy and the convergence of ABC, PSO, and ABC with the forward-communication strategy. The test functions are listed in Eq. (28) to Eq. (30) as follows.

$$f_1(X) = \sum_i^n (\sum_j^i x_j)^2 \quad (28)$$

$$f_2(X) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (29)$$

$$f_3(X) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1 \quad (30)$$

The goal of the optimization is to minimize the outcome. The initial ranges for the test functions are listed in Table 1. For all algorithms, the total population size is 100 and the dimension of the solution space is 100. Each algorithm is executed with 5000 iterations and is repeated with 30 runs. The final result is obtained by taking the average of the outcomes from all runs. The weighting factor of PSO is set to be linearly decreased from 0.9 to 0.4, $c_1 = c_2 = 2$, $r_1, r_2 \in [0, 1]$ are random variables in Eq. (4).

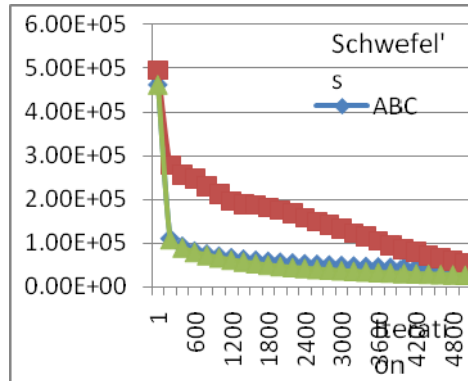
Table 1. Initial ranges for the test functions

Function	Initial range
f_1	$-65.536 \leq x_i \leq 65.536$
f_2	$-5.12 \leq x_i \leq 5.12$
f_3	$-600 \leq x_i \leq 600$

Table 2. The time consumption for obtaining the near best solutions

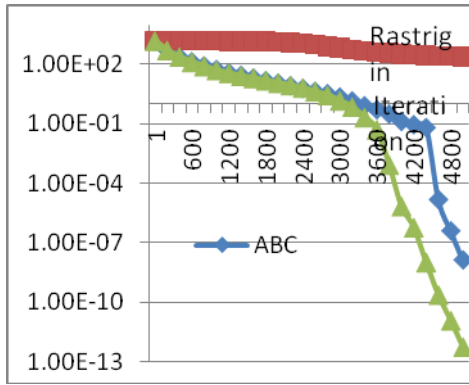
Function		ABC	PSO	ABC-FC
f_1	Fitness value	4.267×10^4	5.389×10^4	2.800×10^4
	Time cost (Sec.)	45.08	32.20	44.57
f_2	Fitness value	1.286×10^{-8}	1.286×10^{-8}	5.125×10^{-13}
	Time cost (Sec.)	8.448	1.215	8.466
f_3	Fitness value	2.319×10^{-15}	5.877×10^{-3}	3.593×10^{-16}
	Time cost (Sec.)	15.99	15.48	14.28

For ABC with forward-communication strategy, $G = 4$, $R_1 = 100$, and $k = 5$ are used. The experimental results are presented in Fig. 6 and the time cost for finding the near best solutions are listed in Table 2. Comparing to the results obtained by PSO and ABC, the accuracy of ABC-FC improves 83% and 73% comparatively. The time cost of finding the near best solutions is reduced about 4% than ABC.

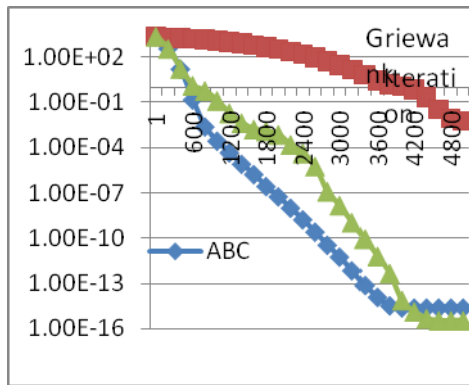


(a)

Fig. 6. Experimental results of the test functions. (a) Schwefel's function. (b) Rastrigin function. (c) Griewank function



(b)



(c)

Fig. 6. (continued)

5 Conclusions

In this paper, we have introduced an overview of several popular algorithms for swarm optimization. Enhancements of corresponding algorithms are also presented, and they are verified by simulations with the traveling salesman problem and popularly employed test functions. We observe that there might be analogies and relationships in distributed computing and social insects. Therefore, developments of new algorithms to a variety of applications of swarm intelligence must be interesting research topics for further studies.

References

1. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999)
2. Bonabeau, E.: Swarm Intelligence. In: O'Reilly Emerging Technology Conference (2003)

3. Eberhart, R., Kennedy, J.: A New Optimizer Using Particle Swarm Theory. In: Proceedings of the Sixth International Symposium on Micro machine Human Science, pp. 39–43. IEEE Press, New York (1995)
4. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of 1995 IEEE International Conf. on Neural Networks, pp. 1942–1948. IEEE Press, New York (1995)
5. Hu, J., Wang, Z., Qiao, S., Gan, J.C.: The Fitness Evaluation Strategy in Particle Swarm Optimization. *Applied Mathematics and Computation* 217, 8655–8670 (2011)
6. Colomi, A., Dorigo, M., Maniezzo, V.: Distributed Optimization by Ant Colonies. In: Varela, F., Bourgine, P. (eds.) *First Eur. Conference Artificial Life*, pp. 134–142 (1991)
7. Dorigo, M., Maniezzo, V., Colomi, A.: Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 26, 29–41 (1996)
8. Dorigo, J.M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation* 1, 53–66 (1997)
9. Chu, S.C., Roddick, J.F., Pan, J.S.: Ant Colony System with Communication Strategies. *Information Sciences* 167, 63–76 (2004)
10. Karaboga, D.: An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report-TR06, Erciyes University, Computer Engineering Department (2005)
11. Passino, K.M.: Biomimicry of Bacterial Foraging for Distributed Optimization and Control. *IEEE Control Systems Magazine* 22, 52–67 (2002)
12. 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)
13. Chu, S.C., Tsai, P.W.: Computational Intelligence Based on the Behavior of Cats. *International Journal of Innovative Computing, Information and Control* 3, 163–173 (2007)
14. Bishop, J.M.: Stochastic Searching Networks. In: *Proc. 1st IEE Conf. on Artificial Neural Networks*, London, pp. 329–331 (1989)
15. Chang, J.F., Chu, S.C., Roddick, J.F., Pan, J.S.: A Parallel Particle Swarm Optimization Algorithm with Communication Strategies. *Journal of Information Science and Engineering* 21, 809–818 (2005)
16. Tsai, P.W., Luo, R., Pan, S.T., Pan, J.S., Liao, B.Y.: Artificial Bee Colony with Forward-communication Strategy. *ICIC Express Letters* 4, 1–6 (2010)

Neural Network Committees Optimized with Evolutionary Methods for Steel Temperature Control

Mirosław Kordos, Marcin Blachnik, Tadeusz Wiczorek, and Sławomir Golak

¹ University of Bielsko-Biała, Department of Mathematics and Informatics,
Bielsko-Biała, Willowa 2, Poland

mkordos@ath.bielsko.pl

² Silesian University of Technology, Department of Management and Informatics,
Katowice, Krasynskiego 8, Poland

marcin.blachnik@polsl.pl

Abstract. This paper presents regression models based on an ensemble of neural networks trained on different data that negotiate the final decision using an optimization approach based on an evolutionary approach. The model is designed for big and complex datasets. First, the data is clustered in a hierarchical way and then using different level of cluster and random choice of training vectors several MLP networks are trained. At the test phase, each network predicts an output for the test vector and the final output is determined by weighing outputs of particular networks. The weights of the outputs are determined by an algorithm based on a merge of genetic programming and searching for the error minimum in some directions. The system was used for prediction the steel temperature in the electric arc furnace in order to shorten and decrease the costs of the steel production cycle.

Keywords: neural network committee, evolutionary algorithms, regression, metallurgy, electric arc furnace.

1 Introduction

In this paper we present a system we used for predicting the steel temperature in the electric arc furnace (EAF) in one of polish steelworks. The purpose of the prediction was to make the steel production cycle shorter and thus cheaper. The first part of the introduction discusses the problems of optimizing steel production in the electric arc furnace and the second part the topic of building committees of predictive models that can be used for complex and large datasets, such as the data obtained from the EAF process.

1.1 EAF Process

In the electric arc furnace the steel scrap is melted using the electric arc to generate most of the heat. Additional heat is obtained from gas that is inserted

and burnt in the furnace. The optimal temperature of the melted steel that is to be tapped out from the furnace is about 1900K, however it must be kept at proper temperature enough long so that all the solid metal melts. If the heating lasts too short not all metal gets melted and if too long, unnecessary time and energy is used and additional wear of the furnace is caused. Besides melting the scrap, the other purpose of the EAF process is to roughly optimize the chemical composition of the steel, before the melted steel goes to next step in the production cycle; the ladle arc furnace, where its chemical composition gets precisely adjusted.

Temperature measurement and control are critical factors in the operation of a modern electric arc furnace. Temperature is measured a few times during every melt. It is done via special lance with thermocouple that dips into the liquid steel. However, every measurement takes time, approximately one minute. The arc has to be turn off and the process suspended. Modern EAFs have the "tap to tap" times even as short as 30 minutes, older ones up to one hour. Waste of time for two, three or even more measurements is thus quite significant.

There are many problems with the continuous measurement of the steel temperature. The temperatures are very high and the radiant heat from the furnace creates major problems for the measuring equipment. Another problem to overcome is the effect of electro-magnetic radiation on sensitive electrical systems within the pyrometer. The conventional temperature measurement systems do not lead to the required solutions. The available systems suffer from numerous application problems that affect the possibility of their practical use.

Millman [9] presented a camera-based technology for monitoring the scrap melting process in an EAF. However, after about two to three minutes of arcing, generation of high dust density had a significantly deleterious effect on the resolution of the images. In addition to dust, the EAF process also generates flames which are composed of a variety of combustion gases and each of them absorbs light to different degrees and at different wavelengths. Therefore, it is not possible to see through a flame, using a camera system that operates under white light. Recently Kendall [6] presented the first stages in the development of a temperature measuring system, which shows the promise of being able to accurately measure the temperature of the steel in the furnace for the whole furnace cycle. However, still many technical problems have to be solved, before the system can be widely used.

Instead of measuring important parameters of the EAF process they can be calculated basing on quantitative time-dependent measurements of energy and mass inputs of the modeled meltdown process [14]. The model allows assessing of the actual mean temperature of liquid steel. Using the model and temperature measured 10 minutes before the tap the melt temperature is calculated and the end of the meltdown phase is predicted. The model is able to predict the tapping temperature. However, we need to know the temperature much earlier, to be able to optimize the parameters during the whole process. Therefore there was a need to build a temperature prediction system that would allow us to limit the number of temperature measurements and thus shorten the EAF process.

We previously built a system based on a single regression model [15,16], as a part of the whole intelligent system for steel production optimization [17]. However, as our recent experiments showed, a significant improvement can be obtained with a committee of regression models. The system we have built using the evolutionary optimized committee and the obtained results are presented in the following chapters.

1.2 Committees of Predictive Models

The dependencies in complex data, like in the metallurgical problem considered here are difficult to be well mapped by a single regression model. There are two ways to improve the prediction of the model: first to split the large datasets into several subsets and second to use an assembly of predictive models. However, there are some problems connected with each of the two approaches. There are usually no clear rules about how to split the data, what to do with the points close to the split boundaries and how far the data may be split before overfitting occurs.

The basic idea behind building a committee of neural networks (or other regression models) is to improve the accuracy of single networks (or other single models). In the case of classification the committee would vote for the final decision, while in the case of regression, the final decision is obtained as a weighted sum of the outputs of particular networks, where the sum of weights equals one.

As it is known from the literature [12], the bias of the whole committee is of the same level as biases of particular networks, while the variance of the committee is smaller or at least not larger than variances of particular networks. For that reason particular networks should have low bias. Low bias can be obtained when the networks are trained without weight regularization. That is quite opposite to a single network, which must keep a bias-variance trade off and some regularization would be beneficial. The reason for which the variance of the committee can be lower is that the correlation between outputs of particular networks is smaller than one. Therefore, the errors sums up in a lower degree than normal arithmetic addition.

Finally a committee can reduce both bias (by using single networks with a small bias) and variance (by summing the variance out).

To further improve the performance of a committee Breiman introduced bagging (bootstrap aggregation) [3]. The idea behind bagging is that to provide decorrelation between the predictions of committee members; each of them should be trained on a different dataset. In bagging the datasets for each network are generated by randomly drawing with replacement n data points from the original data set. The final results are then obtained with a simple voting with equal weights of each network. Experiments showed that bagging tends to work better than training each committee member on the same data. Several variants of that approach exist, e.g. boosting, which implements cascade models, when the vectors on which the previous network makes higher errors are given as the inputs of the next network [10]. AdaBoost (adaptive boosting) [3] is a combination of boosting and bagging, where the probability of selecting a vec-

tor depends on the error the previous network made on that vector (the higher error on previous network, the greater probability of including that vector into the next network input dataset). In the case of big datasets, it may be easier to build a reliable model, dividing the whole data into several small parts, where a different neural network is responsible for modeling each region of the input space. Based on that assumption Jacobs [5] created a model known as Mixture of (local) Experts. The model was further developed into a hierarchical mixture of experts (HMEs), which has at least two layers of gating networks.

Barbosa [2] used a population of feed-forward neural networks, which was evolved using the Clonal Selection Algorithm and the final ensemble was composed of the selected subset of the neural network population.

Recently Chen and Yao [4] proposed a neural network ensemble learning algorithm called Negative Correlation Learning, which introduces a correlation penalty term to the cost function of each network so that each network minimizes not only its mean-square-error but also the correlation with other networks. They used an evolutionary multiobjective algorithm with crossover and mutation operators to optimize the networks in the ensemble. The solution we develop takes advantage of the properties of bagging, Mixture of Experts and evolutionary approach and adds some specific dependencies of the dataset to find the optimal weights of each network in the committee, that can be different for different test vectors.

2 Methodology

2.1 Data Acquisition and Preprocessing

During the EAF process over 100 parameters are measured every second and they are recorded into a database. There are such parameters, as the assessed sorts of metal scraps (which cannot be precisely measured, only assessed by the operator), the mass of the metal scrap, the electric energy used by the electrodes, the amount of gas and oxygen inserted to the steel bath, temperature at several points on the outside surface of the furnace and many others. Thus, the original dataset is very large and the first step is to limit its size by replacing the measurements made with one second frequency with only a limited number of measurements that make sense. The next step is to transform the data obtained at different time stamps into vectors suitable for neural network training. We also had to decide which attributes should be taken as they are (e.g. mass of the metal scrap) and which should be used in the differential form by subtracting from the actual value the value at the time of a previous temperature measurement in order to obtain the change of the values (e.g. temperature, electric energy). Thus we do not predict the temperature of the first measurement, because there is nothing to subtract it from. After performing the two steps we reduced the database size to about 10.000 vectors (on average 3 vectors per one EAF process).

Then the data was standardized and transformed by the hyperbolic tangent function to smoothly limit the influence of outliers [1] and make the data distribution rather closer to a uniform than to Gaussian distribution, which based

on our previous experience described in [17] improves the prediction results for that kind of data.

The next step was to perform feature selection. Feature selection can be either performed as the first stage and then the whole dataset may be split into several subsets, or the split may be done first and feature selection can be performed on each subset independently. A lot of feature selection methods can be used. However, in the model described in this paper we decide to simply remove from the training set the attributes, which had the correlation with output between -0.06 and $+0.06$. This decision was made to remove the feature selection aspect from the discussion in order to better focus on the main topic of this paper. The dataset in the reduced and standardized form with the names of the original variables changed to x_1, x_2, \dots is available from www.kordos.com/datasets/temperature.zip. In this form, the data is still fully usable for machine learning purposes, while it is impossible to discover the technological secrets of the process.

2.2 Data Split and Neural Network Training

As in the Mixture of Expert approach, the whole dataset was split into several parts. We experimented with different clustering methods as k-means, average linkage and fuzzy c-means to divide the training set. Finally we decided to use the average linkage hierarchical clustering with Euclidean distance to divide the data at four different levels:

1. the whole training dataset
2. 3 clusters
3. 9 clusters
4. 27 clusters

The rationale behind the 4-level hierarchy was to take into consideration local properties of the input space using the smaller clusters and still prevent the data overfitting by using also the bigger clusters. Although in hierarchical clustering, as average linkage one does not define a priori the number of clusters, the merging of smaller clusters into bigger ones can be stopped when a predefined average distance between the point pairs is reached and the distance can be dynamically controlled so that the desired number of clusters is obtained. After the clusters were determined, the next step was to prepare the training sets for particular neural networks, which did not exactly cover the clusters. Because the clusters had crisp boundaries and the neural networks should smoothly cover the entire input space, the idea of cluster membership function was used. A point that belongs to a cluster was said to have a membership function value of one and the points of other clusters had the membership value m given by the equation:

$$m = \max(1, 0.5 \cdot c/d) \quad (1)$$

where c is the average distance of points in the current cluster to the cluster center and d is the distance between the given point and the cluster center.

Then the points were sorted according to their m value and as many other cluster points (with the highest m) as the number of points in a given cluster were added to the training set. However, during the network training, the error the network made on such a point was multiplied by the m value of the point.

In this way the ideas of local experts was realized. In order to obtain five decorrelated local networks we implemented the simple bagging algorithm [3]. Although the AdaBoost algorithm [3] is believed to perform on average better, this is not the case with very noisy data or data with a lot of outliers, as it was with our data. Each of the five networks were trained on randomly selected with replacement $2/3$ of the total number of vectors in the local subset. In this way we obtained an ensemble of $5 \cdot (1+3+9+27)=200$ neural networks.

For the MLP network training we used the VSS algorithm [8], however also other MLP training algorithms, as Scaled Conjugate Gradient or Levenberg-Marguardt can be used. The number of hidden layer neurons for each network was randomly chosen from 8 to 16 (based on the experiments this range of hidden layer neurons seemed optimal). The hidden and the output layer neurons had hyperbolic tangent transfer function. In the case of the output layer neuron this transfer function was chosen to limit the outliers' influence on the results, as we described in [17]. However, the MSE we report in the experiment section are based on the original standardized data, that is after transposing the network output by an area tanh function.

At the test phase, first, the distances of the test vector to the cluster centers are calculated. Then only a limited number of the neural networks trained on the clusters that are closer to the test vector take part in the temperature prediction; the networks trained on the whole dataset, all networks from the 3-cluster level, the networks of six of the 9-cluster level and the networks of nine of the 27-cluster level. Thus, together $5 \cdot (1+3+6+9)=95$ neural networks predict the output value for each test vector. Then the networks of each level cluster that take part in the prediction are sorted according to the vector distance to their corresponding cluster centers and their results are saved to an array in the following order: first the whole data set networks, then the 3-cluster level networks starting from the cluster closest to the test vector, then the 9 and 27-cluster networks, each time starting from the cluster closest to the test vector. The outputs of all five networks that cover the whole data and only the average value of the values predicted by the 5 networks of each cluster are saved, thus the arrays has $5+3+6+9=23$ entries and the distances of the vector to particular cluster center are saved in the same order in another 23-entry array. (For the details see Fig (II)).

2.3 Determining the final Decision of Neural Network Committee

First the networks are trained, then they predict the output for each test vector. That is done only once. Only the procedure of determining the weights by which the output of particular networks will be multiplied is iterative, what makes the process relatively fast, because it does not require either re-training or re-testing the neural networks.

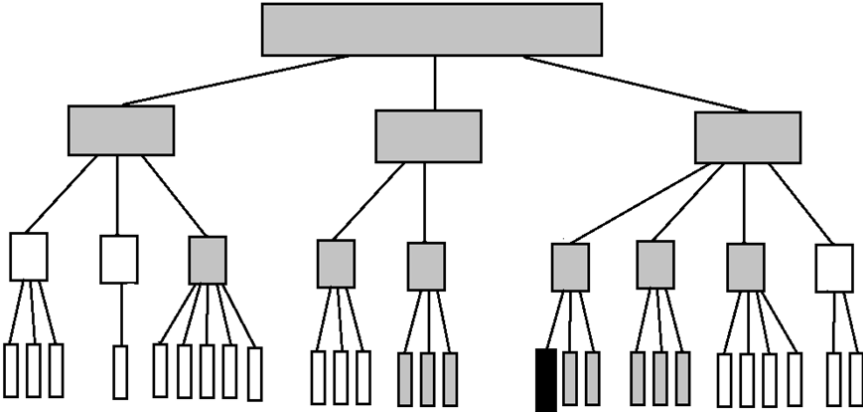


Fig. 1. The average linkage clustering hierarchy. The cluster center closest to the test vector is marked in black. The networks of the clusters participating in the prediction are shaded in grey.

To find the weights w , the following algorithm is used:

1. Create a population of N individuals of randomly generated 23-entry array of weights from the interval $(0, 1)$.
2. When summing the weighted network outputs, multiply each weight by a value inversely proportional to the cluster order on the list plus 1, that is by $1/2$ the weight of the closest cluster, by $1/3$ the weight of the second closest cluster, etc. in each cluster level. Though the step may be omitted, it significantly improves the convergence of the algorithm.
3. Calculate the quality (MSE) of each solution over all test vectors using the following formula for the predicted output of every vector:

$$y_{p,v} = \frac{1}{\sum_i w_i} \sum_i w_i \cdot y_{i,v} \quad MSE = \sqrt{\frac{1}{N} \sum_{v=1}^N (y_{a,v} - y_{p,v})^2} \quad (2)$$

where $y_{p,v}$ is the output predicted by the model for the v -th test vector and $y_{a,v}$ is the actual output for the v -th test vector and w_i is the being optimized weight of i -th network.

4. The winning solution is determined and a randomly chosen half of the other solutions is placed in the minimum along the straight lines connecting the solutions with the winner (Fig. (2)). It is an idea similar to the Self Organizing Migrating Algorithm presented in [18,19].
5. Four best solutions are preserved and included in the next iteration.
6. The genetic operations of crossover and mutation are performed on the weight populations to generate the next population (as in a standard genetic algorithm algorithm). Tsustui discussed multi-parent recombination for real-valued genetic algorithms [13] and similarly we use a random number of parents (from two to four) and a random number of crossover points

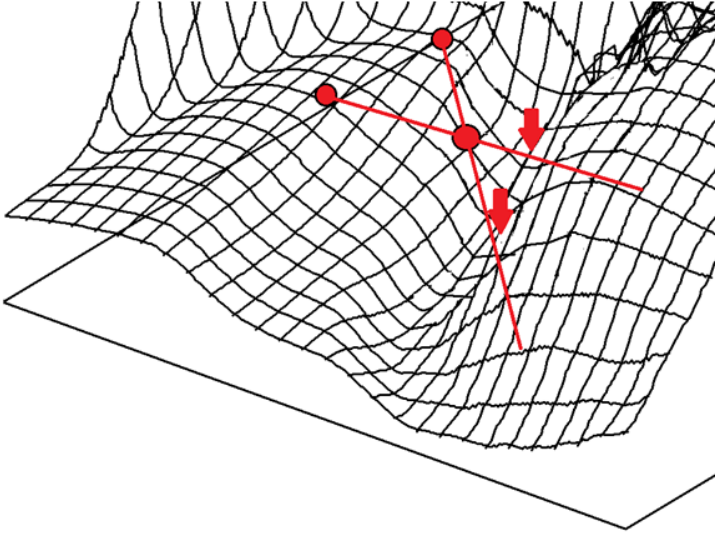


Fig. 2. Moving half of the population to the minimum along the line connecting the points with the winner; to the point shown by the arrows

(from the number of parents minus one to six) to generate each child. Multiply crossover frequently performs better than single crossover, however an optimal number of crossings exists [11], as well as optimal number of parents. MSE in eq. 2 is the value of the fitness function for that individual at the beginning of the training. As the training progresses the fitness function is becoming more and more steep by setting it to higher powers of MSE.

7. Again four best solutions are preserved and included in the next iteration.
8. Repeat steps 3 to 7 as long as a noticeable improvement occurs.

3 Experimental Results

We run the experiments in 10-fold crossvalidation on the dataset that in its differential standardized form as available from www.kordos.com/datasets/temperature.zip. To compare the results with other approaches, we first transposed the data by hyperbolic tangent function and then perform the tests using: only one neural network for the whole training set, using 10 neural networks with average voting and with bagging, using the same cluster schema with the function that minimizes the influence of further cluster networks with average voting and with bagging and our method with local search only (finding minimum in the winner direction) and with evolutionary search only. The results are presented in Table 1. As it could be expected, the quality of the solution improves,

Table 1. Comparison of results obtained with various methods

algorithm	MSE on test set	std. dev.
One network	0.67	0.10
10 networks with average voting	0.62	0.08
10 networks with bagging	0.60	0.08
Avg voting with clusters	0.50	0.06
Bagging with clusters	0.49	0.06
Evolutionary search only	0.48	0.06
Local search only	0.47	0.05
Full method as described in the paper	0.45	0.05

as the additional features are added to the model; committee, bagging, data partitioning, local and global search for the weights of the committee. Another interesting option is the use of hierarchical genetic algorithms [7], where the hierarchy can be composed of the cluster level and the neural networks within particular cluster as the second level. However, we have not attempted this option yet.

4 Conclusions

We presented a hierarchical committee of neural networks and an evolutionary method to optimize the weights of particular committee members. The solution is designed for regression tasks of large and complex datasets. Obviously if the dataset is small, the partitioning into clusters does not make much sense. We applied that system for the prediction of the steel temperature in the electric arc furnace in one of the steelworks in Poland. Previously we implemented an intelligent system to optimize steel production at this steelwork, but the temperature prediction in EAF was realized with single neural network or SVR model and the results were not fully satisfactory. The current solution produces much better outcomes. The dataset depicting the process and the result expected from such a system impose special requirements, for instance optimizing rather the usability of the system than the pure MSE function. That means if some abnormal value is predicted, the MSE of this prediction does not matter, because any abnormal results must be verified by the operator with a temperature measurement. That was one of the reasons, the input and output values are transformed by the hyperbolic tangent function to minimize the system sensitivity to outliers and in order to focus better on typical values.

Currently, the biggest barrier to further improvement of the results is the quality of the data itself, that is the accuracy of the physical value measurements and the fact that sometimes wrong values get recorded by the sensors and frequently finding them is difficult, because it requires taking into account also the sequences of several other values.

Acknowledgement. The project was sponsored by the grant No. 4866/B/T02/2010/38 from the Polish Ministry of Education and Science.

References

1. Angiulli, F., Pizzuti, C.: Fast outlier detection in high dimensional spaces. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002. LNCS (LNAI), vol. 2431, pp. 15–27. Springer, Heidelberg (2002)
2. Barbosa, B., et al.: Evolving an Ensemble of Neural Networks Using Artificial Immune Systems. In: Li, X., Kirley, M., Zhang, M., Green, D., Ciesielski, V., Abbass, H.A., Michalewicz, Z., Hendtlass, T., Deb, K., Tan, K.C., Branke, J., Shi, Y. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 121–130. Springer, Heidelberg (2008)
3. Breiman, L.: Combining predictors. In: Sharkey, A.J.C. (ed.) Combining Artificial Neural Nets, vol. 31. Springer, Heidelberg (1999)
4. Chen, H., Yao, X.: Multiobjective Neural Network Ensembles Based on Regularized Negative Correlation Learning. *IEEE Trans. On Knowledge and Data Engineering* 22, 1738–1751 (2010)
5. Jacobs, R., et al.: Adaptive mixtures of local experts. *Neural Computation* 3, 79 (1991)
6. Kendall, M., et al.: A window into the electric arc furnace, a continuous temperature sensor measuring the complete furnace cycle. *Archives of Metallurgy and Materials* 53(2), 451–454 (2008)
7. Kołodziej, J., et al.: Hierarchical Genetic Computation in Optimal Design. *Journal of Theoretical and Applied Mechanics, "Computational Intelligence"* 42(3), 519–539 (2004)
8. Kordos, M., Duch, W.: Variable step search algorithm for feedforward networks. *Neurocomputing* 71(13–15), 2470–2480 (2008)
9. Millman, M.S., et al.: Direct observation of the melting process in an EAF with a closed slag door. *Archives of Metallurgy and Materials* 53(2), 463–468 (2008)
10. Schapire, R.E.: The strength of weak learnability. *Machine Learning* 5, 197 (1990)
11. Semya, E., et al.: Multiple crossover genetic algorithm for the multiobjective traveling salesman problem. *Electronic Notes in Discrete Mathematics* 36, 939–946 (2010)
12. Tresp, V.: Committee Machines. In: Handbook for Neural Network Signal Processing. CRC Press, Boca Raton (2001)
13. Tsutsui, S., et al.: Multi-parent Recombination with Simplex Crossover in Real Coded Genetic Algorithms. In: Tsutsui, S., et al. (eds.) The 1999 Genetic and Evolutionary Computation Conference, pp. 657–664 (1999)
14. Wendelstorf, J.: Analysis of the EAF operation by process modeling. *Archives of Metallurgy and Materials* 53(2), 385–390 (2008)
15. Wiecek, T.: Intelligent control of the electric-arc steelmaking process using artificial neural networks. *Computer Methods in Material Science* 6(1), 9–14 (2006)
16. Wiecek, T., Blachnik, M., Mączka, K.: Building a model for time reduction of steel scrap meltdown in the electric arc furnace (EAF): General strategy with a comparison of feature selection methods. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 1149–1159. Springer, Heidelberg (2008)
17. Wiecek, T., Kordos, M.: Neural Network-based Prediction of Additives in the Steel Refinement Process. *Computer Methods in Materials Science* 10(1) (March 2010)
18. Zelinka, I., Celikovský, S., Richter, H., Chen, G.: *Evolutionary Algorithms and Chaotic systems*. Springer, Heidelberg (2010)
19. Zelinka, I., Senkerik, R., Oplatková, Z.: Evolutionary Scanning and Neural Network Optimization. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 576–582. Springer, Heidelberg (2008)

Growing Hierarchical Self-Organizing Map for Images Hierarchical Clustering

Bartłomiej M. Buczek and Paweł B. Myszkowski

Applied Informatics Institute, Artificial Intelligence Department
Wrocław University of Technology
Wyb. Wyspiańskiego 27, 51-370 Wrocław, POLAND
bartlomiej.buczek@op.pl, pawel.myszkowski@pwr.wroc.pl
<http://www.ii.pwr.wroc.pl/>

Abstract. This paper presents approaches to hierarchical clustering of images using a GHSOM in application as image search engine. It is analysed some hierarchical clustering and SOMs variants. Experiments are based on benchmark ICPR and MIRFlickr image datasets. As quality of gained solution the external and the internal measures are analysed.

Keywords: Growing Hierarchical Self-Organizing Map, Images Clustering, Hierarchical Images Clustering.

1 Introduction

Clustering as the data mining task can be used in many domain like: recognition, web searching and document segmentation. When it comes to hierarchical clustering it can be used simply anywhere where the hierarchy of data is needed. There are many methods that bases on hierarchical clustering of data for previously artificially created hierarchy by human. There are also some methods, like Growing Hierarchical Self-Organizing Map (GHSOM) which creates hierarchy from scratch.

Images mostly are described in web search engines by the name added to it. The new way of describing document is needed. In [4] there was presented a way for automatic images annotations using GHSOM which can help with creating a new way for image indexing form web search engines. The problem is how to automatically create hierarchy of images. The GHSOM can provide hierarchical structure. It is a matter of research if that structure will be good for images.

The model used in this work will be used in future web search engine. The model of this engine consist on clustering images and assuming this model as a classifier for next images. Hierarchical clustering offers to as a model with many domains which are similar like, for example: in sports - volleyball, football, tennis, etc. - so if the question to model will be sport it can give in return all sports, if volleyball it will return volleyball. Hierarchical clustering because of it structure can provide such mechanism.

This work presents our first research results for quality of clustering for images using GHSOM. In section 2 some types of clustering can be found. Section 3

shows ideas of organizing data as a map. The main idea of this work is included in section 4 and section 5 shows results of our research. Last section concludes and describes directions of further research.

2 Clustering

The data clustering task can be solved by a number of different algorithms for grouping similar objects into respective categories. Clustering is type of unsupervised learning and concerns of many fields, like: machine learning, data mining, pattern recognition, images analysis and information retrieval [5].

There are three main types of clustering: hierarchical clustering, partitioning clustering and spectral clustering [12]. Partitioning clustering generally gives as result in a set of M clusters, which each object belonging to one cluster. Each cluster may be represented by a centroid or a cluster - as this is some sort of summary description of all the objects contained in given cluster. The spectral clustering works by embedding the data points of the partitioning problem into the subspace of the k largest eigenvectors of a normalized affinity / kernel matrix [10].

The hierarchical clustering builds the hierarchy on data set. In this method data are not categorized in respective cluster in one step. A series of categorizations are used, which may begin with single cluster containing all data and ends with n single object clusters. Hierarchical clustering can be distinguished in two ways of creating hierarchy: agglomerative and divisive. The hierarchy can be simply described by a two dimensional diagram known as dendrogram which shows all clusters containing respective subsets of data. The agglomerative hierarchical clustering is based on already existing clusters. This method merges pairs of clusters, however the divisive hierarchical clustering is concerned as an opposite to agglomerative.

3 Organizing Data as a Map

Organizing data was firstly proposed in 1982 by Kohonen to explain the spatial organization of the brain's functions. The data is presented as neural network with the aid of adaptation of weights vectors becomes organized [9].

3.1 Self-Organizing Map

The Self-Organizing Map (SOM) is a computational, unsupervised tool to visualization and analysis of high-dimensional data. There are plenty applications where SOM is used, especially in text mining [9].

The model consists of a number on neural processing elements (commonly called units) where each unit i is assigned an n -dimensional weight vector m_i . The dimension of weight vector is the same as the input patterns dimension (all data vectors have the same dimension as well). The training process of

SOM rely on input pattern presentation and the weight vector adaptation. Each training iteration t begins with selection of input pattern $x(t)$ (often randomly selected). This given pattern is presented to the SOM and each unit determines its activation. The distance between weight vector and input pattern is used to calculate a unit's activation [13]. The units with the smallest distance to input pattern is assigned as Best Matching Unit (*BMU*) of the training iteration:

$$m_{BMU}(t) = \min_i \|x(t) - m_i(t)\| \quad (1)$$

The next and final step is to adapt *BMU* weight vector as weight vectors of *BMU* vicinity units. Geometrically the weight vectors of the adapted units are moved a bit toward the input pattern:

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{BMU_i}(t) \cdot [x(t) - m_i(t)] \quad (2)$$

The amount of weight vector movement is guided by a learning rate α , decreasing in time. The number of units that are affected by adaptation is determined by neighbourhood function, h which also decreases in time [9]. The units in vicinity that are farther from *BMU* are less adapted than the ones closer to the *BMU*.

SOM does not need a target output to be specified. From an initial distribution of random weights, and over many iterations, the SOM eventually settles into a map of stable clusters [9].

3.2 Hierarchical Self-Organizing Map

Interesting way to organize data presents Mäkkulainen [3]. His method can be described as Hierarchical Self-Organizing Map (HSOM) relies on series of SOMs which are placed in layers where the number of SOMs in each structure layer depends on number of SOMs in previous layers (upper one). In this model number of structure layers in HSOM and dimension of maps are defined *a priori*.

In the training process (which always starts from top layers to bottom layer) units weight vectors in SOMs of each layer are adapted. The adapt of weight in next layer is only possible when given layer is finished. Final effect of this process is hierarchical clustering of data set, and approach has some advantages:

- smaller number connections between input and units in HSOM layers [3],
- much shorter processing time which comes from the point above and from hierarchical structure of learning process [3].

In HSOM there are some necessities:

- definition of maps sizes and number of layers which depends on data set,
- choosing of learning parameters for each layer in HSOM.

There is another SOM extension that reduces some above disadvantages.

3.3 Growing Self-Organizing Map

A Growing Self-Organizing Map (GSOM) is another variant of SOM [1] to solve the map size issue. The model consists of number of units which in learning

process grows. The learning process begins with a minimal number of units and grows (by adding new units) on the boundary based on a heuristic. To control the growth of the GSOM there is special value called Spread Factor (SF) - a factor that controls the size of growth. At the beginning of learning process all units are boundary units which means that each unit has the freedom to grow in its own direction. If the unit is chosen to grow it grows in all its free neighbouring positions. However in HSOM still exists a problem of *a priori* given architecture. Another approach, GHSOM solves it.

3.4 Growing Hierarchical Self-Organizing Map

As mentioned before one of the shortcomings of the SOM lies with its fixed architecture that has to be defined *a priori*. Dynamically growing variants of the SOM tends to produce huge maps that are hard to handle [1]. There is another approach that combines advantages of HSOM and GHSOM - the Growing Hierarchical Self-Organizing Map (GHSOM). Architecture of GHSOM allows to grow both in a hierarchical and in a horizontal direction [2]. This provides a convenient structure of clustering for large data sets which can be simple navigate through all layers from top to bottom.

The learning process begins with a virtual map which consist on one unit with weight vector initialized as average of all input data [4]. Also for this unit error (distance) between its weight vector and all data is calculated - this error is global stop parameter in GHSOM learning process. Then next layer map (usually initialised by 2x2 [7]) is created below the virtual layer (this newly created SOM indeed is a child for unit in virtual layer). From now on each map grows in size to represent a collection of data at a specified level of detail. The main difference between growing in GHSOM from GSOM is that a whole row or column of units is added to present SOM layer. The algorithm can be described as:

1. For present SOM start learning process and finish it after l -iterations.
2. For each unit count error (distance) between its weight vector m_i and input patterns $x(t)$ mapped onto this unit in the SOM learning process (this error is called quantization error qe).
3. Check if the sum of quantization errors is greater or equal than certain fraction of quantization error of parent unit:

$$\sum_{n=1}^k qe_i \geq \tau_1 \cdot qe_{parent \ unit}$$

- 3a. If yes - select unit with biggest error and find neighbour to this unit which is most dissimilar. Go to step 4.
else - stop.

4. Between these two units insert a row or a column.
5. Reset learning and neighbour functions for next SOM learning process. Go to step 1.

When the growth process of layer is finished the units of this map are examined for hierarchical expansion [11]. Basically units with large quantization error will add a new SOM for the next layer in the GHSOM structure according to τ_1 value. More specifically when quantization error of unit in examined map is greater than fraction τ_2 of global stop parameter then a new SOM (usually initialised by 2x2) is added to structure. The training process and unit insertion now continues with the newly established SOMs. The difference in learning process of new layer is that fraction of data set is used for training corresponding to units mapped onto parent unit [2].

The whole training process of the GHSOM is terminated when no more units is required for further expansion. This learning process does not necessarily lead to a balanced hierarchy (a hierarchy with equal depth in each branch). To summarize, the growth process of the GHSOM is guided by two parameters τ_1 and τ_2 . The parameter τ_1 controls the growth process in layer (certain fraction in algorithm for expanding in layer for GHSOM) and the parameter τ_2 control hierarchical growth of GHSOM.

4 Images Clustering with GHSOM

The clustering of images is needed because when today's search engine is asked about images usually as an answer comes images which was named by the asked phrase. The better would be when an answer will come on the base of what image contains. The content of images can be described by vector of features - a vector of numbers. GHSOM in simply way bases on vectors of numbers so images can be successfully be used as a data set. In this work the idea was to create a GHSOM structure which will hierarchically organize images in groups of similar images. The visual aspects of such clustering (visualization of images in cluster) can be analysed by human or by some quality measures and ratings.

5 Experiments

We developed two experiments to show the GHSOM application to image clustering. Measures of quality of clusters based on keywords and internal quality of clusters based on distances between units in structure. It is used two benchmark images datasets to examine GHSOM as image searching engine tool effectiveness.

5.1 Used Data Sets - ICPR and MIRFlickr

ICPR data set [15] contains 1109 images segmented to 5x5 grid of 11 features vectors (each image has 275 feature vector) and annotated with 433 keywords.

MIRFlickr-25000 collection [16] consists of 25000 images [8], converted to 5x5 grid of 47 features vectors (each image is represented by 1175 features vector) annotated by annotated by 38 keywords.

Table 1. Confusion matrix representation

		Unit	
		No Keyword	Keyword presented
Image	No keyword	True Negative (TN) = 1	False Positive (FP) = 0
	Keyword presented	False Negative (FN) = -1	True Positive (TP) = 2

5.2 Experiment 1 - External Quality of Clusters

In first experiment the external quality of clusters was examined. For both ICPR and MIRFlickr data set in each experiment 100 images were selected randomly for “test set”. Remaining image set is used as training set for GHSOM. After GHSOM was created and trained each image from “test set” were presented to GHSOM and finer unit were remembered. Then keywords annotated to image were compared with keywords annotated to images (mapped onto unit) creating confusion matrix (see Tab. 1). After the confusion matrix is creation plenty factors can be calculated. In experiment (for quality of clusters) the following factor are being calculated: Accuracy, Precision, Recall and Fmeasure (Fm). Each factor is calculated separately for images and keyword query for both “test sets”. From Tab. 1. factors can be counted as in equations (3), (4), (5), (6).

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP} \quad (3)$$

$$Precision = \frac{TP}{FP + TP} \quad (4)$$

$$Recall = \frac{TP}{FN + TP} \quad (5)$$

$$Fm = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (6)$$

The *Accuracy* can be analysed as value that corresponds to proper results of model in ratio to all data. The *Precision* means how many of the results are correct - so higher value is better. *Recall* explains how many of the positives does model return; so the same - higher is better. *Fm* is harmonic average of *Precision* and *Recall*. Factors for given keyword means that as a query each keyword were used (for ICPR 433 queries, for MIRFlickr 38 queries) and factors for images means that each image was used as a query - for each image keywords were counted as in Tab. 1.

In this experiment some results of *Fm* are some unwanted situations because of *Recall* or *Precision* values equals 0 (occurs division by 0). For each experiment

Table 2. ICPR keyword query: the highest Fm 10 keywords (avg. from 5 runs)

Measure Query	Accuracy	Precision	Recall	Fm
mountain	0,742	0,418	0,797	0,547
Husky	0,902	0,351	0,785	0,473
bushes	0,718	0,280	0,780	0,409
Stadium	0,900	0,295	0,628	0,398
people	0,468	0,262	0,765	0,387
field	0,894	0,254	0,547	0,340
football	0,892	0,246	0,547	0,335
building	0,810	0,239	0,571	0,323
the	0,894	0,235	0,485	0,310
lake	0,780	0,203	0,684	0,306
Average	0,800	0,278	0,659	0,383

Table 3. ICPR image query: the highest Fm 10 images (avg. from 5 runs)

Measure Query	Accuracy	Precision	Recall	Fm
football/image24.jpg	0,988	0,882	0,833	0,857
cannonbeach/image11.jpg	0,993	0,875	0,778	0,824
yellowstone/image36.jpg	0,991	0,750	0,900	0,818
football/image46.jpg	0,979	0,842	0,727	0,780
cannonbeach/image21.jpg	0,991	0,700	0,875	0,778
swissmountains/image21.jpg	0,993	0,625	1,000	0,769
greenland/greenland_168.gif	0,991	0,600	1,000	0,750
football/image45.jpg	0,972	0,762	0,696	0,727
springflowers/image26.jpg	0,991	0,556	1,000	0,714
campusinfall/image10.jpg	0,988	0,750	0,667	0,706
Average	0,988	0,734	0,848	0,772

10 best matches is present but some of results for Fm returns errors because sometimes keywords annotated to image are missing in answer unit keywords. In experiments units are identified as clusters - because each unit has its images mapped onto it in training process. We used the same parameters $\tau_1=0.02$ and $\tau_2=0.05$ values for both data sets.

Experiments for ICPR (see results in Tab. 2. and Tab. 3) shows that experimentally established values of parameters of GHSOM works in the matter of classifying keywords. It means that for image annotated by keywords as a question images with the same keywords can come as an answer from GHSOM. However in ICPR there is plenty badly classified images (almost 40%) if we consider keywords. That's because this data set is not well annotated and it is not well described. However it can be seen external quality of some clusters is high.

Results of experiments for MIRFlickr data set (see Tab. 4. and Tab. 5) shows that for large data sets which are very well described the external quality of

Table 4. MIRFlickr keyword query: the highest Fm 10 keywords (avg. from 5 runs)

Measure \ Query	Accuracy	Precision	Recall	Fm
people	0,468	0,466	0,996	0,633
structures	0,412	0,400	0,980	0,567
plant_life	0,414	0,386	0,974	0,550
people_r1	0,360	0,350	0,992	0,516
indoor	0,402	0,345	0,910	0,499
sky	0,422	0,340	0,883	0,489
female	0,452	0,323	0,799	0,453
male	0,408	0,271	0,779	0,400
night	0,832	0,295	0,589	0,382
tree	0,706	0,306	0,400	0,341
Average	0,488	0,348	0,830	0,483

Table 5. MIRFlickr image query: the highest Fm 10 images (avg. from 5 runs)

Measure \ Query	Accuracy	Precision	Recall	Fm
im13192.jpg	0,947	1,000	0,800	0,889
im5371.jpg	0,921	0,875	0,778	0,824
im1520.jpg	0,921	0,750	0,857	0,800
im20804.jpg	0,921	0,750	0,857	0,800
im6204.jpg	0,921	0,750	0,857	0,800
im14351.jpg	0,921	0,625	1,000	0,769
im20754.jpg	0,921	0,625	1,000	0,769
im24207.jpg	0,921	0,625	1,000	0,769
im3254.jpg	0,921	0,625	1,000	0,769
im7711.jpg	0,921	0,625	1,000	0,769
Average	0,924	0,725	0,915	0,796

cluster is very good. For this data set there is only few errors. From 100 images from “test set” nearly to 10% images were incorrect classified. As it can be seen even for 10 best matches GHSOM for MIRFlickr data set has better clusters than GHSOM for ICPR data set in case of external quality of clusters.

It is worth to mention that keywords with higher Fm were more often used in selected images. Images that has the higher Fm are much better classified if keywords are taken into consideration.

5.3 Experiment 2 - Internal Quality of Clusters

In given experiment Davies Bouldin Index (DBI) and Dunn Index (DI) [6] are used. DBI means that the clustering algorithm that produces a collection of clusters with the smallest index is considered as the best algorithm. Definition of DI concludes that large values of the index indicate the presence of compact and

Table 6. Internal quality of clusters for ICPR and MIRFlickr data sets

		ICPR		MIRFlickr	
τ_1	τ_2	DBI	DI	DBI	DI
0,08	0,05	2,963	0,133	6,170	0,039
0,08	0,01	2,627	0,169	5,900	0,044
0,08	0,005	2,091	0,168	4,949	0,047
0,05	0,05	2,682	0,155	5,616	0,061
0,05	0,01	2,488	0,169	5,292	0,048
0,05	0,005	1,997	0,167	4,863	0,057
0,03	0,05	3,132	0,139	5,895	0,042
0,03	0,01	2,245	0,177	4,652	0,055
0,03	0,005	1,896	0,148	4,781	0,043
0,02	0,005	1,748	0,158	4,179	0,052
0,01	0,005	1,652	0,155	3,166	0,045
0,008	0,004	1,507	0,127	2,988	0,036

well-separated clusters - so if DI is larger then classification is better. For each of data sets quality of clusters was measured for various τ_1 and τ_2 parameters value. For ICPR data set the best results of GHSOM (see Tab. 6.) is given when $\tau_1=0.008$ and $\tau_2=0.004$ were for DBI - generally for smaller both τ_1 and τ_2 the results are better. This can be explained by less images included in each cluster as well as slightly different structure of GHSOM but such small τ_1 and τ_2 leads to very small amount of images in nodes (1 to 3) so its not the best solution. On the other DI scored the best for $\tau_1=0.03$ and $\tau_2=0.01$, which means that for this index better are GHSOM with larger layers but not so large hierarchy.

In MIRFlickr data set experiments (see Tab. 6.) similar conclusion can be found - if smaller τ_1 and τ_2 then better score in DBI but again it implements small amount of images in nodes. But for DI there is no simple way to find best result and in this case the best DI results was observed for $\tau_1=0,05$ and $\tau_2=0,05$ values. DBI for GHSOM is not good quality measure of clusters - better scores always came with small amount of images in nodes which is not the best solution. On the other hand it is hard to explain what τ_1 and τ_2 should be used to find the best DI.

6 Conclusions and the Further Research

In this work GHSOM is presented an effective model of hierarchical image clustering and application as image search engine. Clustering effectiveness was tested by the external and internal quality of clusters. As it can be seen GHSOM can be used for clustering large data sets (as MIRFlickr has 25000 images) and it can be successfully applied to images. There are some badly created clusters and badly classified images but the percent of them are respectively small. For future visual aspects of images in hierarchical clusters created by GHSOM can be researched.

As well as this work only euclidean distance were used so also experiments with different measures for distance should be applied. This work mainly was proceed for future development of document search engine based on images.

Acknowledgements. This work is partially financed from the Ministry of Science and Higher Education Republic of Poland resources in 2010-2013 years as a research SYNAT project (System Nauki i Techniki) in INFINITI-PASSIM.

References

1. Alahakoon, D., Halgamuge, S.K., Sirinivasan, B.: A Self Growing Cluster Development Approach to Data Mining. In: Proc. of IEEE Inter. Conf. on Systems, Man and Cybernetics (1998)
2. Bizzil, S., Harrison, R.F., Lerner, D.N.: The Growing Hierarchical Self-Organizing Map (GHSOM) for analysing multi-dimensional stream habitat datasets. In: 18th World IMACS/MODSIM Congress (2009)
3. Blackmore, J., Mückulainen, R.: Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map. In: Proc. of the IEEE Inter. Conf. on Neural Networks (1993)
4. Chih-Hsiang, C., Chung-Hong, L., Hsin-Chang, Y.: Automatic Image Annotation Using GHSOM. In: Fourth Inter. Conf. on Innovative Comp. Infor. and Control (2009)
5. Fritzke, B.: Some Competitive Learning Methods, Technical Report, Institute for Neural Computation Ruhr-Universität Bochum (1997)
6. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Clustering validity checking methods: part II. ACM SIGMOD Record 31(3) (2002)
7. Herbert, J.P., Yao, J.T.: Growing Hierarchical Self-Organizing Maps for Web Mining. In: Proc. of the 2007 IEEE/WIC/ACM Inter. Confere. on Web Intel. (2007)
8. Huiskes, M.J., Lew, M.S.: The MIR Flickr Retrieval Evaluation. In: ACM Inter. Conf. on Multimedia Inf. Retrieval (2008)
9. Kohonen, T.: Self-organizing maps. Springer, Berlin (1995)
10. Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4) (2007)
11. Rauber, A., Merkl, D., Dittenbach, M.: The GHSOM: Exploratory Analysis of High-Dimensional Data. IEEE Trans. on Neural Networks (2002)
12. Vicente, D., Vellido, A.: Review of Hierarchical Models for Data Clustering and Visualization. In: Girdlez, R., et al. (eds.) Tendencias de la Minera de Datos en Espaa, Espoala de Minera de Datos (2004)
13. Experiments with GHSOM, <http://www.ifs.tuwien.ac.at/~andi/ghsom/experiments.html>
14. Hierarchical clustering, http://www.aiaccess.net/English/Glossaries/GlosMod/e_gm_hierarchical_clustering.htm
15. ICPR data set, <http://www.cs.washington.edu/research/>
16. The MIRFlickr Retrieval Evaluation, <http://press.liacs.nl/mirflickr/>

AdaBoost Ensemble of DCOG Rough–Neuro–Fuzzy Systems

Marcin Korytkowski^{1,2}, Robert Nowicki¹, Leszek Rutkowski^{1,3}, and Rafał Scherer¹

¹ Department of Computer Engineering, Częstochowa University of Technology
al. Armii Krajowej 36, 42-200 Częstochowa, Poland

<http://kik.pcz.pl>

² Olsztyn Academy of Computer Science and Management
ul. Artyleryjska 3c, 10-165 Olsztyn, Poland

<http://www.owskiiz.edu.pl/>

³ Academy of Management (SWSPiZ), Institute of Information Technology,
ul. Sienkiewicza 9, 90-113 Łódź, Poland

marcink@kik.pcz.czest.pl, robert.nowicki@kik.pcz.pl,
lrutko@kik.pcz.czest.pl, rafal@ieee.org

<http://www.swspiz.pl/>

Abstract. Neural networks are able to perfectly fit to data and fuzzy logic systems use interpretable knowledge. These methods cannot handle data with missing or unknown features what can be achieved easily using rough set theory. In the paper we incorporate the rough set theory to ensembles of neuro–fuzzy systems to achieve better classification accuracy. The ensemble is created by the AdaBoost metalearning algorithm. Our approach results in accurate classification systems which can work when the number of available features is changing. Moreover, our rough–neuro–fuzzy systems use knowledge comprised in the form of fuzzy rules to perform classification. Simulations showed very clearly the accuracy of the system and the ability to work when the number of available features decreases.

Keywords: computational intelligence, classifier ensembles, rough sets, fuzzy systems.

1 Introduction

One of soft computing applications is classification which consists in assigning an object described by a set of features to a class. The object $x \in \mathbf{X}$ is described by the vector of features $\mathbf{v} \in \mathbf{V}$. Thus we can equate object x class membership with its feature values $\bar{\mathbf{v}} = [\bar{v}_1, \bar{v}_2, \dots, \bar{v}_n]$ class membership. Consequently, we can use interchangeably x or $\bar{\mathbf{v}}$. Let us assume that fuzzy set $A \subseteq \mathbf{V}$ is given as its membership function $\mu_A(x) = \mu_A(\bar{\mathbf{v}}) = \mu_A(\bar{v}_1, \bar{v}_2, \dots, \bar{v}_n)$ where $\bar{v}_i \in \mathbf{V}_i$ for $i = 1, \dots, n$. We also define the set of all object x features $Q = \{v_1, v_2, \dots, v_n\}$. There are many methods for classifying data. The traditional statistical classification procedures [2] apply Bayesian decision theory and assume knowledge of the posterior probabilities. Unfortunately, in practical situations we have no information about an underlying probability model and

Bayes formula cannot be applied. Over the years, numerous classification methods were developed [2] based on neural networks, fuzzy systems [4], [6], [9], [21], support vector machines, rough sets and other soft computing techniques. These methods do not need the information about the probability model. Yet, they usually fail to classify correctly in case of missing data (features). Generally, there are two ways to solve the problem of missing data:

- Imputation - the unknown values are replaced by estimated ones. The estimated value can be set as the mean of known values of the same feature in other instances. An another idea is to apply the nearest neighbor algorithm based on instances with known value of the same feature. The statistical method can be also used.
- Marginalisation - the features with unknown values are ignored. In this way the problem comes down to the classification in lower-dimensional feature space.

In contrast with the above methods, techniques that use rough sets in fuzzy reasoning [10], [11], [12] do not require to supplement missing features, which process merely masks shortcomings leading to increasing the number of wrong results.

Fuzzy classifiers are frequently used thanks to their ability to use knowledge in the form of intelligible IF-THEN fuzzy rules. Unfortunately neuro-fuzzy systems are not able to cope with missing data. Here rough set systems show their advantage of coping with missing data. They describe the uncertainty of an object classification taking into consideration limited knowledge about the object.

Classifiers can be combined to improve accuracy [5]. By combining intelligent learning systems, the model robustness and accuracy is nearly always improved, comparing to single-model solutions. Popular methods are bagging and boosting which are meta-algorithms for learning different classifiers. They assign weights to learning samples according to their performance on earlier classifiers in the ensemble. Thus subsystems are trained with different datasets created from the base dataset.

In this paper we will combine fuzzy methods with the rough set theory [15], [16], [17] and classifier ensemble methods. An ensemble of neuro-fuzzy systems is trained with the AdaBoost algorithm and the backpropagation. Then rules from neuro-fuzzy systems constituting the ensemble are used in a neuro-fuzzy rough classifier. In this way, we obtain rules that are perfectly fitted to data and use them in the classifier which can operate on data with missing features.

2 Fuzzy Classifiers

Fuzzy classifiers are build out of several functional blocks realizing fuzzification, reasoning, aggregation and defuzzification and rule database. In the paper we restrict to fuzzy systems with singleton fuzzification, DCOG defuzzification, four basic reasoning methods and associated with them appropriate aggregation methods. Fuzzy decision systems built using DCOG defuzzification were proposed in [13], [18], [19], and rough neuro-fuzzy systems in [12]. In this paper, they will be used in a new, modified form, suited for the AdaBoost metalearning. The fuzzy implications which are used in fuzzy classifiers belong to one of the following four groups:

- S-implications

$$I(a, b) = S(N(a), b), \quad (1)$$

which examples can be Łukasiewicz, Reichenbach, Kleene-Dienes, Fodor and Dubois-Prade implications.

- R-implications

$$I(a, b) = \sup_{z \in [0,1]} \{z | T(a, z) \leq b\}, \quad (2)$$

which examples are Rescher, Goguena and Gödel implications.

- QL-implications

$$I(a, b) = S(N(a), T(a, b)), \quad (3)$$

with Zadeh implication

- D-implications [7]

$$I(a, b) = S(T(N(a), N(b)), b). \quad (4)$$

The above definitions of fuzzy implication belongs to so called logical approach. In an alternative, so called Mamdani approach, instead of fuzzy implication t-norms are used. As defined in above mentioned papers, when we use the singleton fuzzification, DCOG defuzzification and Mamdani approach to reasoning we have the following description of output value of fuzzy classifier

$$\bar{z}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot \prod_{k=1}^N T(\mu_{A^k}(\mathbf{v}), \mu_{B_j^k}(\bar{z}_j^r))}{\sum_{r=1}^N \prod_{k=1}^N T(\mu_{A^k}(\mathbf{v}), \mu_{B_j^k}(\bar{z}_j^r))}. \quad (5)$$

When we use the DCOG defuzzification and logical approach to reasoning, the output value is given by

$$\bar{z}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot \prod_{k=1}^N I(\mu_{A^k}(\mathbf{v}), \mu_{B_j^k}(\bar{z}_j^r))}{\sum_{r=1}^N \prod_{k=1}^N I(\mu_{A^k}(\mathbf{v}), \mu_{B_j^k}(\bar{z}_j^r))}. \quad (6)$$

Aggregations methods realised by t-conorm in case of Mamdani approach and t-norm in case of logical approach are incorporated in equations (5) and (6) respectively. As we use the AdaBoost algorithm described later, we made the following assumption regarding centers of fuzzy sets in consequent part of rules:

$$\bar{z}_j^r = \begin{cases} 1 & \text{gdy } x \in \omega_j \\ -1 & \text{gdy } x \notin \omega_j. \end{cases} \quad (7)$$

Moreover, we assume that the sets are defined that

$$\mu_{B_j^r}(z_j) = \begin{cases} 1 & \text{if } z_j = \bar{z}_j^r \\ 0 & \text{if } z_j = -\bar{z}_j^r, \end{cases} \quad (8)$$

Taking into consideration that the centers of consequent fuzzy sets take only two values (assumption (7)), equation (8) can be also written as

$$\mu_{B_j^r}(z_j) = \begin{cases} 1 & \text{if } z_j = \bar{z}_j^r \\ 0 & \text{if } z_j \neq \bar{z}_j^r, \end{cases} \quad (9)$$

or

$$\mu_{B_j^r}(z_j) = z_j \overset{\bar{}}{*} \bar{z}_j^r \quad (10)$$

where operator $\overset{\bar{}}{*}$ is defined as follows

$$a \overset{\bar{}}{*} b = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases} \quad (11)$$

Moreover, for future use, we define the operator $\overset{\neq}{*}$

$$a \overset{\neq}{*} b = \begin{cases} 1 & \text{if } a \neq b \\ 0 & \text{if } a = b \end{cases} \quad (12)$$

Using above assumption we can simplify particular groups of fuzzy implication, in form existing in (6), i.e. $a = \mu_{A^k}(\mathbf{v})$ and $b = \mu_{B_j^k}(\bar{z}_j^r)$. Here are step-by-step transformations:

– S-implication

$$I\left(\mu_{A^k}(\mathbf{v}), \mu_{B_j^k}(\bar{z}_j^r)\right) = N\left(T\left(\mu_{A^k}(\mathbf{v}), \bar{z}_j^k \overset{\neq}{*} \bar{z}_j^r\right)\right), \quad (13)$$

Because of the properties of t-norm, its kind does not matter.

– R-implications

$$I\left(\mu_{A^k}(\mathbf{v}), \mu_{B_j^k}(\bar{z}_j^r)\right) = \left(\mu_{A^k}(\mathbf{v}) \overset{\bar{}}{*} 0\right) \cdot \left(\bar{z}_j^k \overset{\neq}{*} \bar{z}_j^r\right) + \left(\bar{z}_j^k \overset{\bar{}}{*} \bar{z}_j^r\right), \quad (14)$$

Let us note that expression $\mu_{A^k}(\mathbf{v}) \overset{\bar{}}{*} 0$ is a specific case of negation $N(\mu_{A^k}(\mathbf{v}))$. So it is a special case of S-implication.

– QL-implications

$$I\left(\mu_{A^k}(\mathbf{v}), \mu_{B_j^k}(\bar{z}_j^r)\right) = S\left(N(\mu_{A^k}(\mathbf{v})), \mu_{A^k}(\mathbf{v}) \cdot \left(\bar{z}_j^k \overset{\bar{}}{*} \bar{z}_j^r\right)\right), \quad (15)$$

– D-implications We obtain the same expression as in the S-implication case — see eq. (13).

So the systems with D-implication will be identical to system with S-implication. In similar manner we can obtain

$$T\left(\mu_{A^k}(\mathbf{v}), \mu_{B_j^k}(\bar{z}_j^r)\right) = \left(\bar{z}_j^k \overset{\bar{}}{*} \bar{z}_j^r\right) \cdot \mu_{A^k}(\mathbf{v}) \quad (16)$$

for Mamdani approach. Now, let us replace forms of I obtained above to (6) and T to (5). For Mamdani approach we obtain

$$\bar{z}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot S_{k=1}^N \left[\left(\bar{z}_j^k \overset{\bar{}}{*} \bar{z}_j^r\right) \cdot \mu_{A^k}(\mathbf{v}) \right]}{\sum_{r=1}^N \sum_{k=1}^N \left[\left(\bar{z}_j^k \overset{\bar{}}{*} \bar{z}_j^r\right) \cdot \mu_{A^k}(\mathbf{v}) \right]}. \quad (17)$$

For logical approach and S and D-implications we get

$$\bar{z}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot \frac{N}{T} \left[N \left(T \left(\mu_{A^k}(\mathbf{v}), \bar{z}_j^k \neq \bar{z}_j^r \right) \right) \right]}{\sum_{r=1}^N \frac{N}{T} \left[N \left(T \left(\mu_{A^k}(\mathbf{v}), \bar{z}_j^k \neq \bar{z}_j^r \right) \right) \right]}. \quad (18)$$

For R-implications we obtain

$$\bar{z}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot \frac{N}{T} \left[\left(\mu_{A^k}(\mathbf{v}) \bar{*} 0 \right) \cdot \left(\bar{z}_j^k \neq \bar{z}_j^r \right) + \left(\bar{z}_j^k \bar{*} \bar{z}_j^r \right) \right]}{\sum_{r=1}^N \frac{N}{T} \left[\left(\mu_{A^k}(\mathbf{v}) \bar{*} 0 \right) \cdot \left(\bar{z}_j^k \neq \bar{z}_j^r \right) + \left(\bar{z}_j^k \bar{*} \bar{z}_j^r \right) \right]}. \quad (19)$$

For QL-implication we have

$$\bar{z}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot \frac{N}{T} S \left(N \left(\mu_{A^k}(\mathbf{v}), \left(\mu_{A^k}(\mathbf{v}) \cdot \left(\bar{z}_j^k \bar{*} \bar{z}_j^r \right) \right) \right) \right)}{\sum_{r=1}^N \frac{N}{T} S \left(N \left(\mu_{A^k}(\mathbf{v}), \left(\mu_{A^k}(\mathbf{v}) \cdot \left(\bar{z}_j^k \bar{*} \bar{z}_j^r \right) \right) \right) \right)}. \quad (20)$$

3 Rough Fuzzy Classifiers

When we apply to above systems the methodology presented in [10], [11] and [12] we can define the rough fuzzy classifiers, which are capable to work with missing data and are modified to be trained by the AdaBoost metaalgorithm. For Mamdani approach we have

$$\bar{z}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot \frac{N}{S} \left[\left(\bar{z}_j^k \bar{*} \bar{z}_j^r \right) \cdot \mu_{A_L^k}(\mathbf{v}) \right]}{\sum_{r=1}^N \frac{N}{S} \left[\left(\bar{z}_j^k \bar{*} \bar{z}_j^r \right) \cdot \mu_{A_L^k}(\mathbf{v}) \right]}. \quad (21)$$

$$\bar{z}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot \frac{N}{S} \left[\left(\bar{z}_j^k \bar{*} \bar{z}_j^r \right) \cdot \mu_{A_U^k}(\mathbf{v}) \right]}{\sum_{r=1}^N \frac{N}{S} \left[\left(\bar{z}_j^k \bar{*} \bar{z}_j^r \right) \cdot \mu_{A_U^k}(\mathbf{v}) \right]}. \quad (22)$$

where (as is proofed in [12])

$$A_L^r = \begin{cases} \widetilde{P}A^r & \text{gdy } \bar{z}_j^r = 1 \\ \widetilde{P}A^r & \text{gdy } \bar{z}_j^r = -1. \end{cases} \quad (23)$$

and

$$A_U^r = \begin{cases} \widetilde{P}A^r & \text{gdy } \bar{z}_j^r = 1 \\ \widetilde{P}A^r & \text{gdy } \bar{z}_j^r = -1. \end{cases} \quad (24)$$

where the pair $(\widetilde{P}A^r, \overline{P}A^r)$ is the rough fuzzy set defined in the form proposed in [10], [11], and [12]

$$\mu_{\widetilde{P}A^r}(x) = T \left(T_{i:v_i \in P} \mu_{A_i^r}(v_i), T_{i:v_i \in G} \inf_{v_i \in V_i} \mu_{A_i^r}(v_i) \right), \quad (25)$$

$$\mu_{\overline{P}A^r}(x) = T \left(T_{i:v_i \in P} \mu_{A_i^r}(v_i), T_{i:v_i \in G} \sup_{v_i \in V_i} \mu_{A_i^r}(v_i) \right). \quad (26)$$

It takes into consideration vector of only known values of input features i.e. $v_i \in P$ and the infimum and supremum membership of unknown features, i.e. $v_i \in G$.

For logical approach and S and D–implication, we have

$$\underline{\bar{z}}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot T_{k=1}^N \left[N \left(T \left(\mu_{A_L^k}(\mathbf{v}), \bar{z}_j^k \neq \bar{z}_j^r \right) \right) \right]}{\sum_{r=1}^N \sum_{k=1}^N T \left[N \left(T \left(\mu_{A_L^k}(\mathbf{v}), \bar{z}_j^k \neq \bar{z}_j^r \right) \right) \right]}. \quad (27)$$

$$\overline{\bar{z}}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot T_{k=1}^N \left[N \left(T \left(\mu_{A_U^k}(\mathbf{v}), \bar{z}_j^k \neq \bar{z}_j^r \right) \right) \right]}{\sum_{r=1}^N \sum_{k=1}^N T \left[N \left(T \left(\mu_{A_U^k}(\mathbf{v}), \bar{z}_j^k \neq \bar{z}_j^r \right) \right) \right]}. \quad (28)$$

For R–implication

$$\underline{\bar{z}}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot T_{k=1}^N \left[\left(\mu_{A_L^k}(\mathbf{v}) \bar{=} 0 \right) \cdot \left(\bar{z}_j^k \neq \bar{z}_j^r \right) + \left(\bar{z}_j^k \bar{=} \bar{z}_j^r \right) \right]}{\sum_{r=1}^N \sum_{k=1}^N T \left[\left(\mu_{A_L^k}(\mathbf{v}) \bar{=} 0 \right) \cdot \left(\bar{z}_j^k \neq \bar{z}_j^r \right) + \left(\bar{z}_j^k \bar{=} \bar{z}_j^r \right) \right]}. \quad (29)$$

$$\overline{\bar{z}}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot T_{k=1}^N \left[\left(\mu_{A_U^k}(\mathbf{v}) \bar{=} 0 \right) \cdot \left(\bar{z}_j^k \neq \bar{z}_j^r \right) + \left(\bar{z}_j^k \bar{=} \bar{z}_j^r \right) \right]}{\sum_{r=1}^N \sum_{k=1}^N T \left[\left(\mu_{A_U^k}(\mathbf{v}) \bar{=} 0 \right) \cdot \left(\bar{z}_j^k \neq \bar{z}_j^r \right) + \left(\bar{z}_j^k \bar{=} \bar{z}_j^r \right) \right]}. \quad (30)$$

For QL–implication

$$\underline{\bar{z}}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot T_{k=1}^N S \left(N \left(\mu_{A_L^k}(\mathbf{v}), \left(\mu_{A_L^k}(\mathbf{v}) \cdot \left(\bar{z}_j^k \bar{=} \bar{z}_j^r \right) \right) \right) \right)}{\sum_{r=1}^N \sum_{k=1}^N T S \left(N \left(\mu_{A_L^k}(\mathbf{v}), \left(\mu_{A_L^k}(\mathbf{v}) \cdot \left(\bar{z}_j^k \bar{=} \bar{z}_j^r \right) \right) \right) \right)}. \quad (31)$$

$$\overline{\bar{z}}_j = \frac{\sum_{r=1}^N \bar{z}_j^r \cdot T_{k=1}^N S \left(N \left(\mu_{A_U^k}(\mathbf{v}), \left(\mu_{A_U^k}(\mathbf{v}) \cdot \left(\bar{z}_j^k \bar{=} \bar{z}_j^r \right) \right) \right) \right)}{\sum_{r=1}^N \sum_{k=1}^N T S \left(N \left(\mu_{A_U^k}(\mathbf{v}), \left(\mu_{A_U^k}(\mathbf{v}) \cdot \left(\bar{z}_j^k \bar{=} \bar{z}_j^r \right) \right) \right) \right)}. \quad (32)$$

The pair of outputs of the each system, i.e. $(\underline{\bar{z}}_j, \overline{\bar{z}}_j)$ is treated as the interval. When values of all input features are known, there is only one output value $\bar{z}_j = \underline{\bar{z}}_j = \overline{\bar{z}}_j$. When some values are missing, the output value, which is achieved if all features are known, belongs to obtained interval, i.e.

$$\underline{\bar{z}}_j < \bar{z}_j < \overline{\bar{z}}_j \tag{33}$$

The final interpretation of obtained interval, in case of single rough fuzzy classifier is presented in [11], [12], and book [14]. The interpretation for an ensemble is proposed in the next section.

4 Ensembles of DCOG Rough–Neuro–Fuzzy Systems

This section describes ensembles of rough–neuro–fuzzy systems which are created using the AdaBoost. The algorithm which is the most popular boosting method [1], [8], [20], is designed for binary classification. To compute the overall output of the ensemble of classifiers trained by AdaBoost algorithm the following formula is used $f(\mathbf{x}) = \sum_{t=1}^T c_t h_t(\mathbf{x})$, where $c_t = \frac{\alpha_t}{\sum_{t=1}^T \alpha_t}$ is classifier importance for a given training set, $h_t(\mathbf{x})$ is the response of the hypothesis t on the basis of feature vector $\mathbf{x} = [x_1, \dots, x_n]$. The coefficient c_t value is computed on the basis of the classifier error and can be interpreted as the measure of classification accuracy of the given classifier. As we see, the AdaBoost algorithm is a meta-learning algorithm and does not determine the way of learning for classifiers in the ensemble.

$$h_t(x) = \begin{cases} 1 & \text{gdy } \underline{\bar{z}}_j \geq z_{\text{IN}} \text{ i } \overline{\bar{z}}_j > z_{\text{IN}} \\ -1 & \text{gdy } \underline{\bar{z}}_j < z_{\text{OUT}} \text{ i } \overline{\bar{z}}_j \leq z_{\text{OUT}} \\ \frac{1}{2} & \text{gdy } z_{\text{IN}} > \underline{\bar{z}}_j \geq z_{\text{OUT}} \text{ i } \overline{\bar{z}}_j > z_{\text{IN}} \\ -\frac{1}{2} & \text{gdy } \underline{\bar{z}}_j < z_{\text{OUT}} \text{ i } z_{\text{OUT}} < \overline{\bar{z}}_j \leq z_{\text{IN}} \\ 0 & \text{in other cases.} \end{cases} \tag{34}$$

or simpler version

$$h_t(x) = \begin{cases} 1 & \text{gdy } \underline{\bar{z}}_j \geq z_{\text{IN}} \text{ i } \overline{\bar{z}}_j > z_{\text{IN}} \\ -1 & \text{gdy } \underline{\bar{z}}_j < z_{\text{OUT}} \text{ i } \overline{\bar{z}}_j \leq z_{\text{OUT}} \\ 0 & \text{in other cases.} \end{cases} \tag{35}$$

The zero output value of the t -th classifier is interpreted as a refusal to classify because of too small number of features. We used rules from the ensemble of neuro-fuzzy systems (Section 2) in the rough-neuro-fuzzy classifier. If we want to take into account in the overall ensemble response only the classifiers which do not refuse to provide the answer, we redefine aggregation method of all ensemble members into the following form

$$H(x) = \begin{cases} \frac{\sum_{t=1}^T c_t h_t(x)}{\sum_{t=1}^T c_t} & \text{if } \sum_{t=1}^T c_t > 0 \\ 0 & \text{if } \sum_{t=1}^T c_t = 0 \end{cases} \quad (36)$$

$$H(x) = \text{sgn}(f(x)) \quad (37)$$

And the output value can be interpreted as follows

- $H(x) = 1$ — object belongs to the class,
- $H(x) = 0$ — the ensemble does not know,
- $H(x) = -1$ — object does not belong to the class,

when we assume that $z_{\text{IN}} = z_{\text{OUT}} = 0$.

Table 1. Comparison of classifiers performance (WBCD) on learning/testing data set

number of available features	Mamdani approach			S/D-implications		
	correct class. [%]	no class. [%]	incorrect class. [%]	correct class. [%]	no class. [%]	incorrect class. [%]
9	97 / 96	0 / 0	3 / 4	97 / 96	0 / 0	3 / 4
8	91 / 91	8 / 8	1 / 1	91 / 91	8 / 8	1 / 1
7	69 / 70	30 / 30	1 / 1	69 / 70	30 / 30	1 / 1
6	26 / 26	73 / 74	1 / 1	26 / 26	73 / 74	1 / 1
5	20 / 19	80 / 80	0 / 0	20 / 19	80 / 80	0 / 0
4	13 / 12	87 / 87	0 / 0	13 / 12	87 / 87	0 / 0
3	6 / 6	94 / 94	0 / 0	6 / 6	94 / 94	0 / 0
2	2 / 2	98 / 98	0 / 0	2 / 2	98 / 98	0 / 0
1	0 / 0	100 / 100	0 / 0	0 / 0	100 / 100	0 / 0
0	0 / 0	100 / 100	0 / 0	0 / 0	100 / 100	0 / 0

number of available features	R-implications			QL-implications		
	correct class. [%]	no class. [%]	incorrect class. [%]	correct class. [%]	no class. [%]	incorrect class. [%]
9	95 / 92	2 / 3	4 / 6	97 / 96	0 / 0	3 / 4
8	92 / 89	6 / 7	2 / 4	86 / 86	13 / 13	1 / 1
7	88 / 86	10 / 11	2 / 3	45 / 45	55 / 55	1 / 1
6	82 / 80	16 / 17	1 / 3	17 / 17	83 / 83	0 / 0
5	75 / 73	24 / 25	1 / 2	9 / 9	91 / 91	0 / 0
4	65 / 64	34 / 34	1 / 2	3 / 3	97 / 97	0 / 0
3	54 / 53	45 / 46	1 / 1	0 / 0	100 / 100	0 / 0
2	40 / 40	60 / 60	0 / 1	0 / 0	100 / 100	0 / 0
1	23 / 23	77 / 77	0 / 0	0 / 0	100 / 100	0 / 0
0	0 / 0	100 / 100	0 / 0	0 / 0	100 / 100	0 / 0

5 Experimental Results

In this section we test the proposed approach using Wisconsin Breast Cancer Database [3] which consists of instances of binary classes (benign or malignant type of cancer). Classification is based on 9 features (clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, mitoses). The ensemble was created using the AdaBoost algorithm. The fuzzy subsystems were learned by gradient learning and initialized by FCM clustering algorithm. We obtained fuzzy rules from the ensemble and incorporated this rules to the ensemble of neuro-fuzzy rough systems. The classification results for learning and testing datasets are presented in Table II.

6 Conclusions

We obtained fuzzy rules from the ensemble and incorporate this rules to the ensemble of neuro-fuzzy rough systems. The systems have the ability to work on data with missing features. Simulations on Wisconsin Breast Cancer Database benchmark shows very good ability to classify data. Our approach can deal with missing data. As more and more features is missing, the system gradually decreases the number of correct answers (Table II). Classification accuracy was averaged from experiments for every possible combination of missing values.

Acknowledgments. This work was partly supported by the Foundation for Polish Science Team Programme (co-financed by the EU European Regional Development Fund, Operational Program Innovative Economy 2007-2013) and the Polish Ministry of Science and Higher Education (Habilitation Project 2008-2010 Nr N N514 4141 34, Polish-Singapore Research Project 2008-2011).

References

1. Breiman, L.: Bias, variance, and arcing classifiers. Tech. Rep. In: Technical Report 460, Statistics Department, University of California (1997)
2. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience Publication, Hoboken (2000)
3. Frank, A., Asuncion, A.: UCI machine learning repository (2010), <http://archive.ics.uci.edu/ml>
4. Jang, R.J.S., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence. Prentice Hall, Upper Saddle River (1997)
5. Kuncheva, L.: Combining Pattern Classifiers. Studies in Fuzziness and Soft Computing. John Wiley & Sons, Chichester (2004)
6. Kuncheva, L.I.: Fuzzy Classifier Design. Studies in Fuzziness and Soft Computing. Physica-Verlag, A Springer-Verlag Company, Heidelberg (2000)
7. Mas, M., Monserrat, M., Torrens, J.: Two types of implications derived from uninorms. Fuzzy Sets and Systems 158, 2612–2626 (2007)
8. Meir, R., Rätsch, G.: An introduction to boosting and leveraging. In: Mendelson, S., Smola, A.J. (eds.) Advanced Lectures on Machine Learning. LNCS(LNAI), vol. 2600, pp. 118–183. Springer, Heidelberg (2003)

9. Nauck, D.: Foundations of Neuro-Fuzzy Systems. John Wiley, Chichester (1997)
10. Nowicki, R.: On combining neuro-fuzzy architectures with the rough set theory to solve classification problems with incomplete data. *IEEE Trans. Knowl. Data Eng.* 20(9), 1239–1253 (2008), doi:10.1109/TKDE.2008.64
11. Nowicki, R.: Rough-neuro-fuzzy structures for classification with missing data. *IEEE Trans. Syst., Man, Cybern. B* 39(6), 1334–1347 (2009)
12. Nowicki, R.: On classification with missing data using rough-neuro-fuzzy systems. *International Journal of Applied Mathematics and Computer Science* 20(1), 55–67 (2010), doi:10.2478/v10006-010-0004-8
13. Nowicki, R., Rutkowska, D.: Neuro-fuzzy systems based on gödel and sharp implication. In: *Proceedings of Intern. Conference: Application of Fuzzy Systems and Soft Computing (ICAFS-2000)*, pp. 232–237. Siegen, Germany (2000)
14. Nowicki, R.K.: Fuzzy decision systems for tasks with limited knowledge. *Academic Publishing House EXIT* (2009) (in polish)
15. Pawlak, Z.: Rough sets. *International Journal of Information and Computer Science* 11(341), 341–356 (1982)
16. Pawlak, Z.: *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer, Dordrecht (1991)
17. Pawlak, Z.: Rough sets, decision algorithms and bayes' theorem. *European Journal of Operational Research* 136, 181–189 (2002)
18. Rutkowska, D., Nowicki, R.: Implication - based neuro-fuzzy architectures. *International Journal of Applied Mathematics and Computer Science* 10(4), 675–701 (2000)
19. Rutkowska, D., Nowicki, R., Rutkowski, L.: Neuro-fuzzy architectures with various implication operators. In: *The State of the Art in Computational Intelligence - Proc. Intern. Symposium on Computational Intelligence (ISCI 2000)*, Kosice, pp. 214–219 (2000)
20. Schapire, R.E.: A brief introduction to boosting. In: *Conference on Artificial Intelligence*, pp. 1401–1406 (1999)
21. Wang, L.X.: *Adaptive Fuzzy Systems and Control*. PTR Prentice Hall, Englewood Cliffs (1994)

A Two-Armed Bandit Collective for Exemplar Based Mining of Frequent Itemsets with Applications to Intrusion Detection

Vegard Haugland, Marius Kjølleberg,
Svein-Erik Larsen, and Ole-Christoffer Granmo

University of Agder, Grimstad, Norway

Abstract. Over the last decades, frequent itemset mining has become a major area of research, with applications including indexing and similarity search, as well as mining of data streams, web, and software bugs. Although several efficient techniques for generating frequent itemsets with a minimum support (frequency) have been proposed, the number of itemsets produced is in many cases too large for effective usage in real-life applications. Indeed, the problem of deriving frequent itemsets that are both compact and of high quality, remains to a large degree open.

In this paper we address the above problem by posing frequent itemset mining as a collection of interrelated *two-armed bandit* problems. In brief, we seek to find itemsets that frequently appear as subsets in a stream of itemsets, with the frequency being constrained to support granularity requirements. Starting from a randomly or manually selected exemplar itemset, a collective of Tsetlin automata based two-armed bandit players aims to learn which items should be included in the frequent itemset. A novel reinforcement scheme allows the bandit players to learn this in a decentralized and on-line manner by observing one itemset at a time. Since each bandit player learns simply by updating the state of a finite automaton, and since the reinforcement feedback is calculated purely from the present itemset and the corresponding decisions of the bandit players, the resulting memory footprint is minimal. Furthermore, computational complexity grows merely linearly with the cardinality of the exemplar itemset.

The proposed scheme is extensively evaluated using both artificial data as well as data from a real-world network intrusion detection application. The results are conclusive, demonstrating an excellent ability to find frequent itemsets at various level of support. Furthermore, the sets of frequent itemsets produced for network intrusion detection are compact, yet accurately describe the different types of network traffic present.

1 Introduction

Over the last two decades, frequent itemset mining has become a major area of research, with applications including indexing and similarity search, as well as mining of data streams, web, and software bugs [1].

The problem of finding frequent itemsets can be formulated as follows. Consider a set I of n items, $I = \{i_1, i_2, \dots, i_n\}$. A *transaction* T_i , $1 \leq i \leq m$, is defined as a subset of I , $T_i \subseteq I$, collectively referred to as a transaction set: $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$. When an arbitrary set X is a subset of a transaction T_i , $X \subseteq T_i$, one says that T_i supports X . The *support* of X is then simply the number of transactions T_i in \mathcal{T} that supports X , $\text{support}(X) = |\{T_i \in \mathcal{T} | X \subseteq T_i\}|$, with $|\cdot|$ denoting set cardinality. The notion of interest in this paper – the *frequency* of an itemset – can then be defined as follows:

Definition 1 (Itemset Frequency). *The frequency of itemset X , $\text{freq}(X)$, is defined as the fraction of transactions T_i in \mathcal{T} that supports X :*

$$\text{freq}(X) = \frac{|\{T_i \in \mathcal{T} | X \subseteq T_i\}|}{|\mathcal{T}|}.$$

Although several efficient techniques for generating frequent itemsets with a minimum frequency have been proposed [1], the number of itemsets produced is in many cases too large for effective usage in real-life applications. Indeed, the problem of deriving frequent itemsets that are both compact and of high quality, so that they are tailored to perform well in specific real-life applications, remains to a large degree open.

1.1 Our Approach

In this paper we address the above problem by posing frequent itemset mining as a collective intelligence problem, modelled as a collection of interrelated *two-armed bandit* problems. The two-armed bandit problem [5] is a classical optimization problem where a player sequentially pulls one of multiple arms attached to a gambling machine, with each pull resulting in a random reward. The reward distributions are unknown, and thus, one must balance between exploiting existing knowledge about the arms, and obtaining new information.

Our proposed scheme can be summarized as follows. Starting from a randomly or manually selected exemplar transaction, a collective of so-called Tsetlin automata [7] based bandit players – one automaton for each item in the exemplar – aims to learn which items should be included in the mined frequent itemset, and which items should be excluded. A novel reinforcement scheme allows the bandit players to learn this in a decentralized and on-line manner, by observing transactions one at a time, as they appear in the transaction stream. Since each bandit player learns simply by updating the state of a finite automaton, and since the reinforcement feedback is calculated purely from the present transaction and the corresponding decisions of the bandit players, the resulting memory footprint is minimal. Furthermore, computational complexity grows merely linearly with the cardinality of the exemplar transaction.

The above Tsetlin automata based formulation of frequent itemset mining provides us with three distinct advantages:

1. Any desired target itemset frequency can be achieved without any more memory than what is required by the Tsetlin automata in the collective (one byte per automaton).

2. Itemsets are found by the means of on-line collective learning, supporting processing of on-line data streams, such as streams of network packets.
3. An exemplar transaction is used to focus the search towards frequent itemsets that are both compact and of high quality, tailored to perform well in real-life applications.

1.2 Example Application — Network Anomaly Detection

Network intrusion detection has been a particularly promising application area for frequent itemset mining [8,9]. In so-called network anomaly detection, huge amounts of network packet data needs to be mined so that the patterns of normal traffic can be found, and so that anomalous traffic can be distilled as deviations from the identified patterns. Although not based on frequent itemset mining, the packet byte based anomaly detection approach of Mahoney [3] is particularly fascinating in this perspective because it achieves state-of-the-art anomaly detection performance simply by inspecting 48 bytes from the header of network packets.

In order to investigate to what degree the properties of our bandit problem based approach to frequent itemset mining can be taken advantage of in network anomaly detection, we will propose a packet byte based anomaly detection system, formulated as a frequent itemset problem. Informally stated, each network packet i is seen as a transaction T_i and each byte value from the network packet is seen as an item belonging to the transaction. Thus, in this application we are looking for frequent itemsets consisting of byte-value pairs, such as $\{dstaddr1 : 24, dstaddr2 : 34, tcpflag : 12\}$, which is an itemset that identifies network packets with destination 24.34.*.* and tcp-flag 12.

1.3 Paper Organization

The paper is organized as follows. First, in Sect. 2 we present our decentralized Tsetlin automata based solution to frequent itemset mining, as well as a novel reinforcement scheme that guides the collective of Tsetlin automata towards a given target itemset frequency. Then, in Sect. 3 we demonstrate the performance advantages of the introduced scheme, including its ability to robustly identify compact itemsets that are useful for summarizing both artificial as well as real-life data. Finally, in Sect. 4 we offer conclusions as well as pointers to further work.

2 A Collective of Two-Armed Bandit Players for Exemplar Based Frequent Itemset Mining

We here target the problem of finding frequent itemsets with a given support by *on-line* processing of transactions, taking advantage of so-called transaction *exemplars*. To achieve this, we design a collective of Learning Automata (LA) that builds upon the work of Tsetlin and the linear two-action automaton [4,7].

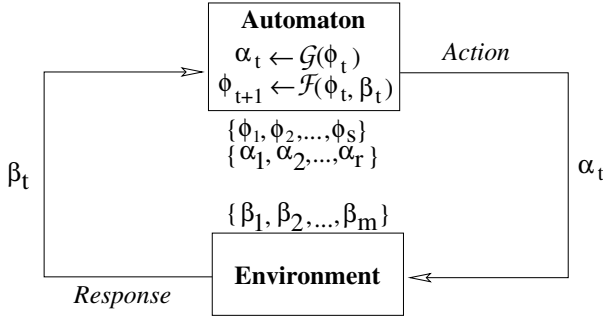


Fig. 1. A Learning Automaton interacting with an Environment

Generally stated, an LA performs a sequence of actions on an *Environment*. The Environment can be seen as a generic *unknown* medium that responds to each action with some sort of reward or penalty, generated *stochastically*. Based on the responses from the Environment, the aim of the LA is to find the action that minimizes the expected number of penalties received. Fig. 1 shows the interaction between a LA and the Environment.

As illustrated in the figure, an LA can be defined in terms of a quintuple [4]:

$$\{\underline{\Phi}, \underline{\alpha}, \underline{\beta}, \mathcal{F}(\cdot, \cdot), \mathcal{G}(\cdot, \cdot)\}.$$

$\underline{\Phi} = \{\phi_1, \phi_2, \dots, \phi_s\}$ is the set of internal automaton states, $\underline{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ is the set of automaton actions, and, $\underline{\beta} = \{\beta_1, \beta_2, \dots, \beta_m\}$ is the set of inputs that can be given to the automaton. An output function $\alpha_t = \mathcal{G}[\phi_t]$ determines the next action performed by the automaton given the current automaton state. Finally, a transition function $\phi_{t+1} = \mathcal{F}[\phi_t, \beta_t]$ determines the new automaton state from the current automaton state as well as the response of the Environment to the action performed by the automaton.

Based on the above generic framework, the crucial issue is to design automata that can learn the optimal action when interacting with the Environment. Several designs have been proposed in the literature, and the reader is referred to [4,6] for an extensive treatment.

2.1 The Item Selector Automaton (ISA)

Our LA based scheme for solving frequent itemset problems is centered around the concept of an *exemplar* transaction $T_E \subset I$. With the exemplar transaction T_E as a basis, the goal of our scheme is to identify an itemset $X \subseteq T_E$ whose frequency, $freq(X)$, is equal to a specific target frequency γ .

At the heart of our scheme we find an Item Selector Automaton (ISA). In brief, for each item i_j in T_E , a dedicated ISA, based on the Tsetlin automaton [7], is constructed, having:

- States: $\underline{\Phi} = \{-N - 1, -N, \dots, -1, 0, \dots, N - 2, N\}$.

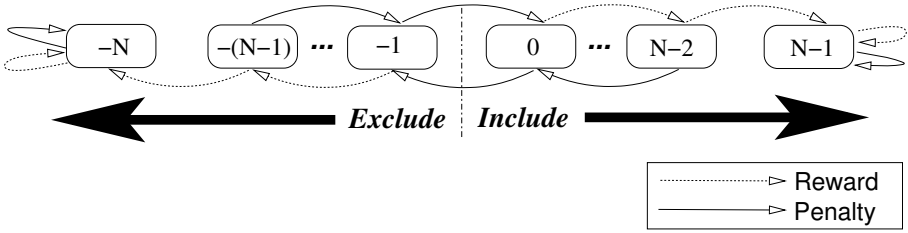


Fig. 2. An ISA choosing between including or excluding an item from candidate frequent itemsets

- Actions: $\underline{\alpha} = \{Include, Exclude\}$.
- Inputs: $\underline{\beta} = \{Reward, Penalty\}$.

Fig. 2 specifies the \mathcal{G} and \mathcal{F} matrices.

The \mathcal{G} matrix can be summarized as follows. If the automaton state is positive, then action *Include* will be chosen by the automaton. If on the other hand the state is negative, then action *Exclude* will be chosen. Note that since we initially do not know which action is optimal, we set the initial state of the ISA randomly to either -1 or 0 .

The state transition matrix \mathcal{F} determines how learning proceeds. As seen in the graph representation of \mathcal{F} found in the figure, providing a *reward* input to the automaton *strengthens* the currently chosen action, essentially by making it less likely that the other action will be chosen in the future. Correspondingly, a *penalty* input *weakens* the currently selected action by making it more likely that the other action will be chosen later on. In other words, the automaton attempts to incorporate past responses when deciding on a sequence of actions.

Note that our ISA described above deviates from the traditional Tsetlin automaton in one important manner: State $-N$ and state $N - 1$ are absorbing. This allows the ISA to converge to a single state, rather than to a distribution over states, thus artificially introducing an unambiguous convergence criterion.

2.2 Reinforcement Scheme

Since each item i_j in the transaction exemplar T_E is assigned a dedicated ISA, ISA_j , we obtain a collective of ISA. The reinforcement scheme presented here is incremental, processing one transaction at a time at discrete time steps. At each time step s , a transaction $T_i \in \mathcal{T}$ is presented to the collective of ISA, whose responsibility is to propose a candidate itemset $X(s)$ for that time step. By on-line processing of the transactions, the goal of the ISA is to converge to proposing an itemset X^* that is supported with frequency, $freq(X^*) = \gamma$, with probability arbitrarily close to 1.

To elaborate, each automaton, ISA_j , chooses between two options at every time step s : shall its own item i_j be included in $X(s)$ or shall it be excluded? Based on the decisions of the ISAs as a collective, a candidate itemset $X(s)$ for

time step s is produced. A response from the Environment is then incurred as follows. First it is checked whether the present transaction T_i supports $X(s)$, and based on the presence or absence of support, each ISA_j is rewarded/penalized according to the following novel reinforcement scheme.

The novel reinforcement scheme that we propose rewards an automaton ISA_j based on the decision of the automaton at time step s and based on whether the present transaction T_i supports the resulting candidate itemset $X(s)$. In brief, if ISA_j decides to include item i_j in $X(s)$, we have two possibilities. If T_i supports $X(s)$, ISA_j is rewarded. On the other hand, if T_i does not support $X(s)$, then ISA_j is randomly penalized with probability $r = \frac{\gamma}{1-\gamma}$. The other decision ISA_j can make is to exclude item i_j from $X(s)$. For that decision, ISA_j is randomly rewarded with probability $r = \frac{\gamma}{1-\gamma}$ if T_i does not support $X(s) \cup \{i_j\}$. On the other hand, if T_i supports $X(s) \cup \{i_j\}$, then the ISA is penalized.

The above reinforcement scheme is designed to guide the collective of learning automata as a whole towards converging to including/excluding items in $X(s)$ so that the frequency of $freq(X(s))$ converges to γ , with probability arbitrarily close to 1.

Note that because multiple variables, and thereby multiple ISA, may be involved when constructing the frequent itemset, we are dealing with a game of LA [4]. That is, multiple ISA interact with the same Environment, and the response of the Environment depends on the actions of several ISA. In fact, because there may be conflicting goals among the ISA involved, the resulting game is competitive. The convergence properties of general competitive games of LA have not yet been successfully analyzed, however, results exist for certain classes of games, such as the Prisoner's Dilemma game [4].

In order to maximize speed of learning, we initialize each ISA randomly to either the state ' 1 ' or ' 0 '. In this initial configuration, the actions will be switched relatively quickly because only a single state transition is necessary for a switch. Accordingly, the joint state space of the ISA is quickly explored in this configuration. However, as learning proceeds and the ISA move towards their boundary states, i.e., states ' N ' and ' $N-1$ ', the exploration calms down. Accordingly, the search for a solution to the frequent itemset problem at hand becomes increasingly focused.

Furthermore, note that we keep a time step counter for each ISA. When a certain cut off threshold has been achieved, we force one of the ISA to converge if it has not yet done so. This enforcement resets the counters of the other ISA, allowing them to adapt to the new configuration. The purpose of this mechanism is to increase convergence speed in ambiguous decision making cases where two different actions provide more or less the same feedback.

3 Empirical Results

In this section we evaluate our proposed scheme using both artificial data as well as data from a real-world network intrusion detection application.

3.1 Artificial Data

For the evaluation on artificial data we constructed a collection of transactions in a manner that by selecting the correct itemset X , one can achieve a frequency, $freq(X)$, of either 0.0, 0.125, 0.25, \dots , 0.75, 0.875, 1.0. The purpose is to challenge the scheme by providing a large number of frequency levels to choose among, with only one of these being the target frequency γ that our scheme must converge to. By varying the target frequency γ in the latter range, we also investigate the robustness of our scheme towards low, medium, and high frequency itemsets.

We here report an ensemble average after conducting 100 runs of our scheme. Given a target frequency γ , each run produces an itemset X^* , with X^* being supported by an actual frequency $freq(X^*)$. By comparing the sought target frequency γ with the achieved frequency $freq(X^*)$, the convergence accuracy of our scheme is revealed.

We first study convergence accuracy when any subset $X \subset T_E$ of the exemplar transaction T_E have a frequency, $freq(X)$, that is either equal to the target frequency γ , unity (1), or zero (0). Then the goal of the ISA collective is to identify the subset $X \subset T_E$ with frequency γ . As seen in Fig. 3, our scheme achieves this goal with remarkable accuracy.

We observe that for any of the target frequencies γ listed in the figure, on average our ISA collective identifies itemsets X^* with frequencies $freq(X^*) \in \{0.0, 0.125, 0.25, \dots, 0.75, 0.875, 1.0\}$ that either equals γ or surpasses γ with the least possible amount: $freq(X^*) \geq \gamma \wedge freq(X^*) - 0.125 < \gamma$.

When using a generic transaction exemplar T_E instead — one that contains item subsets $X \subseteq T_E$ of any arbitrary frequency level $freq(X) \in \{0.0, 0.125, 0.25, \dots, 0.75, 0.875, 1.0\}$, the challenge increases. The ISA collective then also have the option to produce frequencies in close vicinity of the target

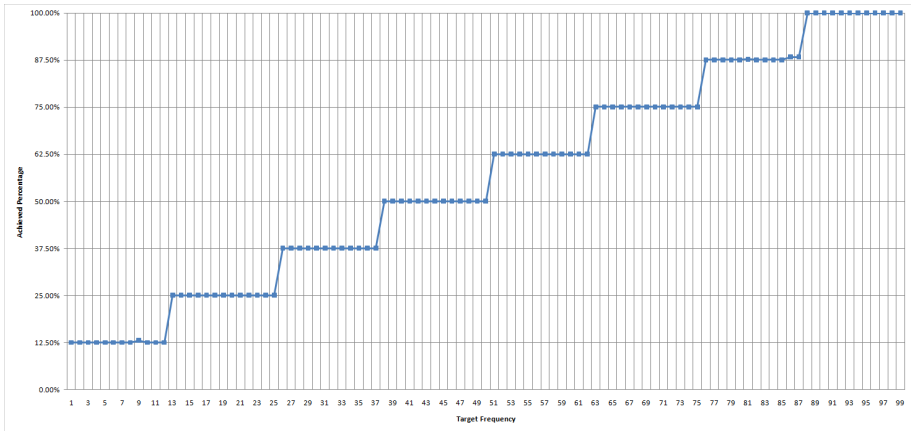


Fig. 3. Achieved percentage of transactions supported by produced itemset (y-axis) using a specific exemplar transaction, for varying target frequencies γ (x-axis)

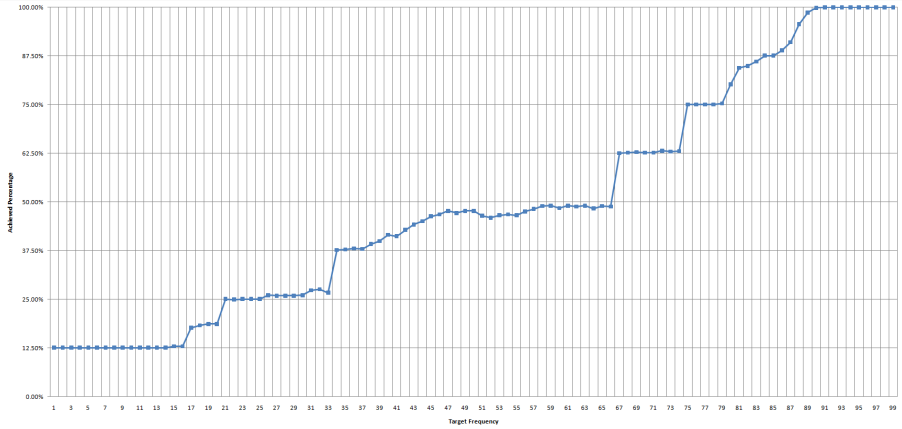


Fig. 4. Achieved percentage of transactions supported by produced itemset (y-axis) using a generic exemplar transaction, for varying target frequencies γ (x-axis)

frequency γ . Fig. 4 reports the resulting convergence accuracy, and as seen, it is now more difficult for the collective of ISA to always produce an itemset X with a transaction support frequency exactly equal to γ . Still, the itemsets produced are always close to a nearby neighbor of γ in $\{0.0, 0.125, 0.25, \dots, 0.75, 0.875, 1.0\}$

3.2 DARPA Intrusion Detection Evaluation Data Set

To evaluate the ISA collective scheme on a real life application, we have implemented a network intrusion detection system, with the ISA collective at its core. Briefly explained, we analyze the last 40 bytes of each network packet header in combination with the first 8 bytes of the transport layer payload, as also done in NETAD [3].¹ Essentially, we see each network packet as a transaction, and byte-value pairs from a network packet are seen as items.

We intend to detect network attacks by first learning a collection of frequent itemsets that describe the key features of normal network traffic – and based on these frequent itemsets, reporting network packets as anomalous when they do not support any of the learned frequent itemsets.

We use the 1999 DARPA Intrusion Detection Evaluation data set [2] for training and testing our system. During training, we use one week of normal traffic data, learning one frequent itemset at a time by randomly picking exemplar transactions (network packets) from the normal traffic data. Each time the collective of ISA converges to a new frequent itemset, all network packets that support this itemset are removed from the normal traffic data, and the procedure is repeated to learn each kind of traffic. Note that the granularity of the learned frequent itemset is controlled by γ .

¹ Note that in contrast to NETAD, we analyze both ingoing and outgoing network packets, for greater accuracy.

Table 1. Transaction Exemplars

Rule	Attacks
ver+ihl:0x45, frag1:0x40, frag2:0x00, proto:0x06, src-port1:0x00, tcphl:0x50, urgptr1:0x00, urgptr2:0x00	ps
ver+ihl:0x45, dscp:0x00, frag1:0x00, frag2:0x00, proto:0x06, tcphl:0x50, urgptr1:0x00, urgptr2:0x00	ps
ver+ihl:0x45, dscp:0x00, len:0x00, frag1:0x40, frag2:0x00, ttl:0x40, proto:0x06, dstaddr1:0xac, dstaddr2:0x10, dstport1:0x00, dstport2:0x17, tcphl:0x50, recwd1:0x7d, recwd2:0x78, urgptr1:0x00, urgptr2:0x00, pld1:0x00, pld5:0x00, pld7:0x00, pld8:0x00, pld9:0x00	ps, guesstelnet, sendmail

For testing, the second week of the DARPA data set is used. The network packets from this week also contain attacks. If a network packet in the second week of data does not support any of the learned frequent itemsets, it is reported as an anomaly. The complete results of these experiments will be reported in a forthcoming paper, however, Table 1 contains a few representative examples of frequent itemsets, called Rules, and which kind of attacks they allow us to detect. As seen, each Rule consists of selected bytes from a packet, combined with a hexadecimal representation of the corresponding byte value. Thus, considering the first row of the table, network packets of the so-called ps-attack do not support the frequent itemset {ver+ihl:0x45, frag1:0x40, frag2:0x00, proto:0x06, src-port1:0x00, tcphl:0x50, urgptr1:0x00, urgptr2:0x00}, and are therefore reported as anomalies.

Finally, when it comes to computational complexity, note that since each bandit player learns simply by updating the state of a finite automaton, and since the reinforcement feedback is calculated purely from the present itemset and the corresponding decisions of the bandit players, the resulting memory footprint is minimal (usually one byte per ISA). Furthermore, computational complexity grows merely linearly with the cardinality of the exemplar itemset.

4 Conclusion

In this paper we have addressed frequent itemset mining by the means of a collective of so-called Item Selector Automata (ISA). By on-line interaction with a stream of transactions, the collective of ISA decides which items should be excluded and which should be included in a frequent itemset, with items being chosen from a randomly or manually selected exemplar itemset. A novel reinforcement scheme guides the ISA towards finding a candidate itemsets that is supported by transactions with a specified frequency.

Since each bandit player learns simply by updating the state of a finite automaton, and since the reinforcement feedback is calculated purely from the present itemset and the corresponding decisions of the bandit players, the resulting memory footprint is minimal. Furthermore, computational complexity grows merely linearly with the cardinality of the exemplar itemset.

In extensive evaluation using both artificial data and data from a real-world network intrusion detection application, we find the results quite conclusive, demonstrating that the ISA collective possesses an excellent ability to find frequent itemsets at various levels of support. Furthermore, the sets of frequent itemsets produced for network intrusion detection are compact, yet accurately describe the different types of network traffic present, allowing us to detect attacks in the form of anomalies.

In our further work, we intend to develop formal convergence proofs for the ISA collective. We are also presently investigating a hierarchical scheme for organizing ISA collectives, with the purpose of increased scalability.

References

1. Han, J., Chen, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 55–86 (2007)
2. Lippmann, R., Haines, J., Fried, D., Korba, J., Das, K.: The 1999 DARPA off-line intrusion detection evaluation. *Computer Networks* 34(4), 579–595 (2000)
3. Mahoney, M.V.: Network traffic anomaly detection based on packet bytes. In: *Proceedings of ACM-SAC 2003*, pp. 346–350. ACM, New York (2003)
4. Narendra, K.S., Thathachar, M.A.L.: *Learning Automata: An Introduction*. Prentice Hall, Englewood Cliffs (1989)
5. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. MIT Press, Cambridge (1998)
6. Thathachar, M.A.L., Sastry, P.S.: *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Kluwer Academic Publishers, Dordrecht (2004)
7. Tsetlin, M.L.: *Automaton Theory and Modeling of Biological Systems*. Academic Press, London (1973)
8. Vaarandi, R., Podins, K.: Network ids alert classification with frequent itemset mining and data clustering. In: *Proceedings of the 2010 IEEE Conference on Network and Service Management*. IEEE, Los Alamitos (2010)
9. Wang, H., Li, Q.H., Xiong, H., Jiang, S.Y.: Mining maximal frequent itemsets for intrusion detection. In: Jin, H., Pan, Y., Xiao, N., Sun, J. (eds.) *GCC 2004*. LNCS, vol. 3252, pp. 422–429. Springer, Heidelberg (2004)

Applications of Paraconsistent Artificial Neural Networks in EEG

Jair Minoro Abe^{1,2}, Helder F.S. Lopes², Kazumi Nakamatsu³, and Seiki Akama⁴

¹ Graduate Program in Production Engineering, ICET - Paulista University

R. Dr. Bacelar, 1212, CEP 04026-002 São Paulo – SP – Brazil

² Institute For Advanced Studies – University of São Paulo, Brazil

jairabe@uol.com.br, helder.mobile@gmail.com

³ School of Human Science and Environment/H.S.E. – University of Hyogo – Japan

nakamatu@shse.u-hyogo.ac.jp

⁴ C-Republic, Tokyo, Japan

akama@jcom.home.ne.jp

Abstract. In this work we summarize all our studies on Paraconsistent Artificial Neural Networks applied to electroencephalography.

Keywords: artificial neural network, paraconsistent logics, EEG analysis, pattern recognition, Dyslexia.

1 Introduction

Generally speaking, Artificial Neural Network (ANN) can be described as a computational system consisting of a set of highly interconnected processing elements, called artificial neurons, which process information as a response to external stimuli. An artificial neuron is a simplistic representation that emulates the signal integration and threshold firing behavior of biological neurons by means of mathematical structures. ANNs are well suited to tackle problems that human beings are good at solving, like prediction and pattern recognition. ANNs have been applied within several branches, among them, in the medical domain for clinical diagnosis, image analysis and interpretation signal analysis and interpretation, and drug development.

So, ANN constitutes an interesting tool for electroencephalogram (EEG) qualitative analysis. On the other hand, in EEG analysis we are faced with imprecise, inconsistent and paracomplete data. In this paper we employ a new kind of ANN based on paraconsistent annotated evidential logic $\mathcal{E}\tau$, which is capable of manipulating imprecise, inconsistent and paracomplete data in order to make a first study of the recognition of EEG standards.

The EEG is a brain electric signal activity register, resultant of the space-time representation of synchronic postsynaptic potentials. The graphic registration of the sign of EEG can be interpreted as voltage flotation with mixture of rhythms, being frequently sinusoidal, ranging from 1 to 70 Hz. In the clinical-physiological practice, such frequencies are grouped in frequency bands as can see in Figure 1.

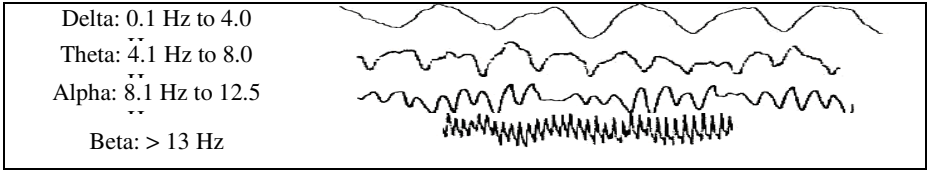


Fig. 1. Frequency bands clinically established and usually found in EEG

2 Background

Paraconsistent Artificial Neural Network (PANN) is a new artificial neural network introduced in [7]. Its basis leans on paraconsistent annotated logic $E\tau$ [1]. Let us present it briefly.

The atomic formulas of the logic $E\tau$ are of the type $p_{(\mu, \lambda)}$, where $(\mu, \lambda) \in [0, 1]^2$ and $[0, 1]$ is the real unitary interval (p denotes a propositional variable). $p_{(\mu, \lambda)}$ can be intuitively read: “It is assumed that p ’s favorable evidence is μ and contrary evidence is λ .” Thus: $p_{(1.0, 0.0)}$ can be read as a true proposition; $p_{(0.0, 1.0)}$ can be read as a false proposition; $p_{(1.0, 1.0)}$ can be read as an inconsistent proposition; $p_{(0.0, 0.0)}$ can be read as a paracomplete (unknown) proposition; $p_{(0.5, 0.5)}$ can be read as an indefinite proposition.

We introduce the following concepts (all considerations are taken with $0 \leq \mu, \lambda \leq 1$): Uncertainty degree (Eq. 2.1) and Certainty degree (Eq. 2.2);

$$G_{un}(\mu, \lambda) = \mu + \lambda - 1 \quad (2.1)$$

$$G_{ce}(\mu, \lambda) = \mu - \lambda \quad (2.2)$$

An order relation is defined on $[0, 1]^2$: $(\mu_1, \lambda_1) \leq (\mu_2, \lambda_2) \Leftrightarrow \mu_1 \leq \mu_2$ and $\lambda_1 \leq \lambda_2$, constituting a lattice that will be symbolized by τ .

With the uncertainty and certainty degrees we can get the following 12 output states (Table 2.1): *extreme states*, and *non-extreme states*.

Table 1. Extreme and Non-extreme states

Extreme states	Symbol	Non-extreme states	Symbol
True	V	Quasi-true tending to Inconsistent	$Qv \rightarrow T$
False	F	Quasi-true tending to Paracomplete	$Qv \rightarrow \perp$
Inconsistent	T	Quasi-false tending to Inconsistent	$Qf \rightarrow T$
Paracomplete	\perp	Quasi-false tending to Paracomplete	$Qf \rightarrow \perp$
		Quasi-inconsistent tending to True	$T \rightarrow v$
		Quasi-inconsistent tending to False	$T \rightarrow f$
		Quasi-paracomplete tending to True	$\perp \rightarrow v$
		Quasi-paracomplete tending to False	$\perp \rightarrow f$

3 The Main Artificial Neural Cells

In the PANN, the certainty degree G_{ce} indicates the ‘measure’ falsity or truth degree. The uncertainty degree G_{un} indicates the ‘measure’ of the inconsistency or paracompleteness. If the certainty degree is low or the uncertainty degree is high, it generates an indefinision.

The resulting certainty degree G_{ce} is obtained as follows:

- If: $V_{icc} \leq G_{un} \leq V_{scc}$ or $V_{scct} \leq G_{un} \leq V_{icct} \Rightarrow G_{ce} = \text{Indefinition}$
- For: $V_{cpa} \leq G_{un} \leq V_{scct}$
 If: $G_{un} \leq V_{icc} \Rightarrow G_{ce} = \text{False with degree } G_{un}$
 Else: $V_{scct} \leq G_{un} \Rightarrow G_{ce} = \text{True with degree } G_{un}$

A Paraconsistent Artificial Neural Cell – PANC – is called *basic* PANC when given a pair (μ, λ) is used as input and resulting as output:

- $S_{2a} = G_{un} =$ resulting uncertainty degree
- $S_{2b} = G_{ce} =$ resulting certainty degree
- $S_1 = X =$ constant of Indefinition.

Using the concepts of *basic* Paraconsistent Artificial Neural Cell , we can obtain the family of PANC considered in this work as described in Table 3.1 below:

Table 2. Paraconsistent Artificial Neural Cells

PANC	Inputs	Calculations	Output
Analytic connection: PANCac	μ λ Ft_{ct} Ft_{ce}	$\lambda_c = 1 - \lambda$ $G_{un} G_{ce}$ $\mu_r = (G_{ce} + 1)/2$	If $ G_{ce} > Ft_{ce}$ then $S_1 = \mu_r$ and $S_2 = 0$ If $ G_{un} > Ft_{ct}$ and $ G_{un} > G_{ce} $ then $S_1 = \mu_r$ and $S_2 = G_{un} $ if not $S_1 = 1/2$ and $S_2 = 0$
Maximization: PANCmax	μ λ	G_{ce} $\mu_r = (G_{ce} + 1)/2$	If $\mu_r > 0.5$, then $S_1 = \mu$ If not $S_1 = \lambda$
Minimization: PANCmin	μ λ	G_{ce} $\mu_r = (G_{ce} + 1)/2$	If $\mu_r < 0.5$, then $S_1 = \mu$ if not $S_1 = \lambda$

4 PANN for Morphological Analysis

The process of morphological analysis of a wave is performed by comparing with a certain set of wave patterns (stored in the control database). A wave is associated with a vector (finite sequence of natural numbers) through digital sampling. This vector characterizes a wave pattern and is registered by PANN. Thus, new waves are compared, allowing their recognition or otherwise.

Each wave of the survey examined the EEG corresponds to a portion of 1 second examination. Every second of the exam contains 256 positions.

The wave that has the highest favorable evidence and lowest contrary evidence is chosen as the more similar wave to the analyzed wave.

A control database is composed by waves presenting 256 positions with perfect sinusoidal morphology, with 0,5 Hz of variance, so taking into account Delta, Theta, Alpha and Beta (of 0.5 Hz to 30.0 Hz) wave groups.

4.1 Data Preparation

The process of wave analysis by PANN consists previously of data capturing, adaptation of the values for screen examination, elimination of the negative cycle and normalization of the values for PANN analysis.

As the actual EEG examination values can vary highly, in module, something 10 μV to 1500 μV , we make a normalization of the values between 100 μV and -100 μV by a simple linear conversion, to facilitate the manipulation the data:

$$x = \frac{100.a}{m} \tag{4.1}$$

Where: m is the maximum value of the exam; a is the current value of the exam.
 x is the current normalized value.

The minimum value of the exam is taken as zero value and the remaining values are translated proportionally. It is worth to observe that the process above does not allow the loss of any wave essential characteristics for our analysis.

4.2 The PANN Architecture

The architecture of the PANN used in decision making is based on the architecture of Paraconsistent Artificial Neural System for Treatment of Contradictions [7].

This method is used primarily for PANN (fig. 4.1) to balance the data received from expert systems. After this process uses a decision-making lattice to determine the soundness of the recognition (fig. 4.2).

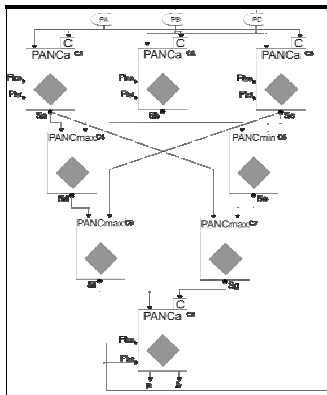


Fig. 2 The architecture for morphological analysis.

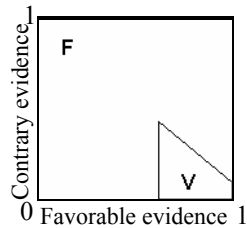


Fig. 3 Lattice for decision-making used in morphological; F: logical state false, interpreted as wave not similar; V: logical state true, interpreted as wave similar.

Table 3. Lattice for decision-making (Fig. 4.2) used in the morphological analysis

Limits of areas of lattice		
True	Fe > 0,61	Ce < 0,40 G _{ce} > 0,22
False	Fe < 0,61	Ce > 0,40 G _{ce} ≤ 0,23

Ce: contrary evidence; Fe: favorable evidence; G_{ce}: certainty degree;

4.1 Expert System 1 – Checking the Number of Wave Peaks

The aim of the *expert system 1* is to compare the waves and analyze their differences regarding the number of peaks.

$$Se_1 = 1 - \left(\frac{|bd - vt|}{bd + vt} \right) \quad (4.2)$$

Where: vt is the number of peaks of the wave; Se_1 is the value for expert system 1.
 bd is the number of peaks of the wave stored in the database.

4.2 Expert System 2 – Checking Similar Points

The aim of the *expert system 2* is to compare the waves and analyze their differences regarding of similar points.

When we analyze the similar points, it means that we are analyzing how one approaches the other point.

$$Se_2 = \frac{\sum_{j=1}^n (x_j)}{n} \quad (4.3)$$

Where: n is the total number of elements; x is the element of the current position.
 j is the current position; Se_2 is the value for expert system 2.

4.3 Expert System 3 – Checking Different Points

The aim of the *expert system 3* is to compare the waves and analyze their differences regarding of different points.

When we analyze the different points, it means that we are analyzing how a point more distant from each other, so the factor of tolerance should also be considered.

$$Se_3 = 1 - \left(\frac{\sum_{j=1}^n \left(\frac{|x_j - y_j|}{a} \right)}{n} \right) \quad (4.4)$$

n is the total number of elements; a is the maximum amount allowed, j is the current position; x is the value of wave 1, y is the value of wave 2; Se_3 is the value for expert system 3.

5 Experimental Procedures – Differentiating Frequency Bands

In our work we've studied two types of waves, specifically delta and theta waves band, where the size of frequency established clinically ranges (Fig. 1.1).

Seven exams of different EEG were analyzed, being two exams belonging to adults without any learning disturbance and five exams belonging to children with learning disturbance[9][17][18].

Each analysis was divided in three rehearsals, each rehearsal consisted of 10 seconds of the analyzed, free from visual analysis of spikes and artifacts regarding to the channels T3 and T4.

In the first battery it was used of a recognition filter belonging to the Delta band. In the second battery it was used a filter for recognition of waves belonging to the Theta band. In the third battery it was not used any filters for recognition.

Table 4. Contingency table

		Visual Analysis					Total
		Delta	Theta	Alpha	Beta	Unrecognized	
PANN Analysis	Delta	31	3	0	0	0	34
	Theta	15	88	1	1	0	105
	Alpha	0	5	22	0	0	27
	Beta	0	0	1	3	0	4
	N/D	7	2	1	0	0	10
Total		53	98	25	4	0	180

Index Kappa = 0.80

Table 5. Statistical results - sensitivity and specificity: Delta waves

		Visual analysis		
		Delta	Not Delta	Total
PANN	True	31	124	155
	False	22	3	25
	Total	53	127	180

Sensitivity = 58%; Specificity = 97%

Table 6. Statistical results - sensitivity and specificity: Theta waves

		Visual analysis		
		Alpha	Not Alpha	Total
PANN	True	22	150	172
	False	3	5	8
	Total	25	155	180

Sensitivity = 88%; Specificity = 96%

Table 7. Statistical results - sensitivity and specificity: Alpha waves

		Visual analysis		
		Theta	Not Theta	Total
PANN	True	88	65	153
	False	10	17	27
	Total	98	82	180

Sensitivity = 89%; Specificity = 79%

Table 8. Statistical results - sensitivity and specificity: Beta waves

		Visual analysis		
		Beta	Not Beta	Total
PANN	True	3	175	178
	False	1	1	2
	Total	4	176	180

Sensitivity = 75%; Specificity = 99%

Table 9. Statistical results - sensitivity and specificity: Unrecognized waves

		Visual analysis		
		Unrecognized	Recognized	Total
PAN	True	0	170	170
	False	0	10	10
N	Total	0	180	180

Sensitivity = 100%; Specificity = 94%

6 Experimental Procedures – Applying in Alzheimer disease

It is known that the visual analysis of EEG patterns may be useful in aiding the diagnosis of AD, and indicated in some clinical protocols for diagnosing the disease [15][16]. The most common findings on visual analysis of EEG patterns are slowing of brain electrical activity based on predominance of delta and theta rhythms and decrease or absence of alpha rhythm. However, these findings are more common and evident in patients in moderate or advanced stages of disease [21][22][23].

In this study we have sixty-seven Analyzed EEG records, thirty-four normals and thirty-three probable AD (p value = 0.8496) during the awake state at rest.

All tests were subjected to morphological analysis methodology for measuring the concentration of waves. Later this information is submitted to an PANN unit responsible for assessing the data and arriving at a classification of the examination in Normal or probable AD (Fig. 6.1).

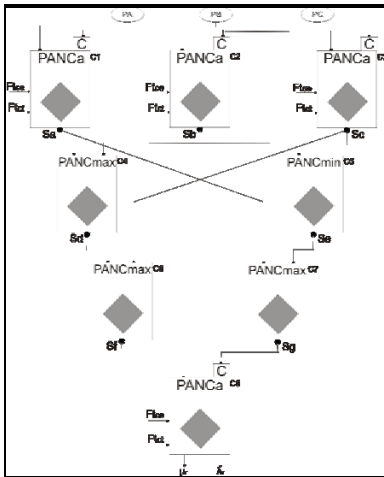


Fig. 4 The architecture for diagnostic analysis

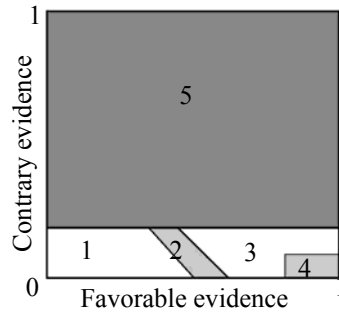


Fig. 5 Lattice for decision-making used in diagnostic analysis (Fig. 6.1). Area 1: State logical False (AD likely below average population), 2: State logical Quasi-true (AD likely than average population); Area 3: State logical Quasi-false (Normal below average population); Area 4: State logical True (Normal above average population); Area 5: logical state of uncertainty (not used in the study area).

Table 10. Lattice for decision-making (Fig. 6.1) used in diagnostic analysis used after making PANN analysis (Fig. 6.2)

Limits of areas of lattice	
Area 1	$G_{ce} \leq 0,1999$ and $G_{ce} \geq 0,5600$ and $ G_{un} < 0,3999$ and $ G_{un} \geq 0,4501$
Area 2	$0,2799 < G_{ce} < 0,5600$ and $0,3099 \leq G_{un} < 0,3999$ and $Fe < 0,5000$
Area 3	$0,1999 < G_{ce} < 0,5600$ and $0,3999 \leq G_{un} < 0,4501$ and $Fe > 0,5000$
Area 4	$G_{ce} > 0,7999$ and $ G_{un} < 0,2000$

Ce: contrary evidence; Fe: favorable evidence; G_{ce} : certainty degree; G_{un} : uncertainty degree;

6.1 Expert System 1 – Detecting the Diminishing Average Frequency Level

The aim of the *expert system 1* is An expert system verifies the average frequency level of Alpha waves and compares them with a fixed external one (external parameter wave).

Such external parameter can be, for instance, the average frequency of a population or the average frequency of the last exam of the patient. This system also generates two outputs: favorable evidence μ (normalized values ranging from 0 (corresponds to 100% – or greater frequency loss) to 1 (which corresponds to 0% of frequency loss) and contrary evidence λ (Eq. 6.1).

The average frequency of population pattern used in this work is 10 Hz.

$$\lambda = 1 - \mu \quad (6.1)$$

6.2 Expert System 2 – High Frequency Band Concentration

The aim of the *expert system 2* is the expert system is utilized for alpha band concentration in the exam. For this, we consider the quotient of the sum of fast alpha and beta waves over slow delta and theta waves (Eq. 7.2) as first output value. For the second output value (contrary evidence λ) is used Eq. 6.1.

$$\mu = \left(\frac{(A+B)}{(D+T)} \right) \quad (6.2)$$

Where: A is the alpha band concentration; B is the beta band concentration.

D is the delta band concentration; T is the theta band concentration.

μ is the value resulting from the calculation.

6.3 Expert System 3 – Low Frequency Band Concentration

The aim of the *expert system 3* is the expert system is utilized for tetha band concentration in the exam. For this, we consider the quotient of the sum of slow delta and theta waves over fast alpha and beta waves (Eq. 6.3) as first output value. For the second output value (contrary evidence λ) is used Eq. 6.1.

$$\mu = \left(\frac{(D+T)}{(A+B)} \right) \tag{6.3}$$

Where: *A* is the alpha band concentration; *B* is the beta band concentration.
D is the delta band concentration; *T* is the theta band concentration.
 μ is the value resulting from the calculation.

6.4 Results

Table 11. Diagnosis – Normal x Probable AD patients

		<i>Gold Standard</i>		
		AD patient	Normal patient	Total
<i>PANN</i>	AD patient	35.82%	14.93%	50.75%
	Normal patient	8.96%	40.30%	49.25%
	Total	44.78%	55.22%	100.00%

Sensitivity = 80%; Specificity = 73%; Index of coincidence (Kappa): 76%

7 Conclusions

We believe that a process of the examination analysis using a PANN attached to EEG findings, such as relations between frequency bandwidth and inter hemispheric coherences, can create computational methodologies that allow the automation of analysis and diagnosis.

These methodologies could be employed as tools to aid in the diagnosis of diseases such as dyslexia or Alzheimer, provided they have defined electroencephalographic findings.

In the case of Alzheimer's disease, for example, in studies carried out previously shown satisfactory results [10] (but still far from being a tool to aid clinical) that demonstrated the computational efficiency of the methodology using a simple morphological analysis (only paraconsistent annotated logic $E\tau$). These results encouraged us to improve the morphological analysis of the waves and try to apply the method in other diseases besides Alzheimer's disease.

With the process of morphological analysis using the PANN, it becomes possible to quantify the frequency average of the individual without losing its temporal reference. This feature becomes a differential, compared to traditional analysis of quantification of frequencies, such as FFT (Fast Fourier Transform), aiming at a future application in real-time analysis, i.e. at the time of acquisition of the EEG exams.

Regarding the specificity, the method showed more reliable results. Taking into account an overall assessment in the sense we take the arithmetic mean of sensitivity (75.50%) and specificity (92.75%), we find reasonable results that encourage us to seek improvements in this study.

Even finding a low sensitivity in the recognition of delta waves, the methodology of pattern recognition using morphological analysis showed to be effective, achieving recognize patterns of waves similar to patterns stored in the database, allowing quantifications and qualifications of the examination of EEG data to be used by PANN in their process analysis of examination.

References

1. Abe, J.M.: Foundations of Annotated Logics, PhD thesis, USP, Brazil (1992) (in Portuguese)
2. Abe, J.M.: Some Aspects of Paraconsistent Systems and Applications. *Logique et Analyse* 157, 83–96 (1997)
3. Abe, J.M., Lopes, H.F.S., Anghinah, R.: Paraconsistent Artificial Neural Network and Alzheimer Disease: A Preliminary Study. *Dement. Neuropsychol.* 3, 241–247 (2007)
4. Anghinah, R.: Estudo da densidade espectral e da coerência do eletrencefalograma em indivíduos adultos normais e com doença de Alzheimer provável, PhD thesis, FMUSP, São Paulo (2003) (in Portuguese)
5. Ansari, D., Karmiloff-Smith, A.: Atypical trajectories of number development: a neuroconstructivist perspective. *Trends In Cognitive Sciences* 12, 511–516 (2002)
6. Blonds, T.A., Attention-Deficit Disorders and Hyperactivity. In *Developmental Disabilities in Infancy and Ramus, F., Developmental dyslexia: specific phonological deficit or general sensorimotor dysfunction? Current Opinion in Neurobiology* 13, 1-7 (2003)
7. Da Silva Filho, J.I., Abe, J.M., Torres, G.L.: *Inteligência Artificial com as Redes de Análises Paraconsistentes LTC-Livros Técnicos e Científicos Editora S.A., São Paulo, 313 pág* (2008) (in Portuguese)
8. Hynd, G.W., Hooper, R., Takahashi, T.: Dyslexia and Language-Based disabilities. In: Brumbak, C. (ed.) *Text Book of Pediatric Neuropsychiatry*, pp. 691–718. American Psychiatric Press, Washington (1985)
9. Lindsay, R.L.: Dyscalculia. In: Capute, Accardo (eds.) *Developmental Disabilities in Infancy and Childhood*, pp. 405–415. Paul Brookes Publishing Co., Baltimore (1996)
10. Lopes, H.F.S.: Aplicação de redes neurais artificiais paraconsistentes como método de auxílio no diagnóstico da doença de Alzheimer, MSc Dissertation, Faculdade de Medicina-USP, São Paulo, p. 473 (2009) (in Portuguese)
11. Klimeshc, W.: EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain Res. Ver.* 29, 169–195 (1999)
12. Klimesch, W., Doppelmayr, H., Wimmer, J., Schwaiger, D., Rôhm, D., Bruber, W., Hutzler, F.: Theta band power changes in normal and dyslexic children. *Clin. Neurophysiol.* 113, 1174–1185 (2001)
13. Kocyigit, Y., Alkan, A., Erol, H.: Classification of EEG Recordings by Using Fast Independent Component Analysis and Artificial Neural Network. *J. Med. Syst.* 32(1), 17–20 (2008)
14. Niedermeyer, E., da Silva, F.L.: *Electroencephalography*, 5th edn. Lippincott Williams & Wilkins (2005)

15. Claus, J.J., Strijers, R.L.M., Jonkman, E.J., Ongerboer De Visser, B.W., Jonker, C., Walstra, G.J.M., Scheltens, P., Gool, W.A.: The diagnostic value of EEG in mild senile Alzheimer's disease. *Clin. Neurophysiol.* 18, 15–23 (1999)
16. Crevel, H., Gool, W.A., Walstra, G.J.M.: Early diagnosis of dementia: Which tests are indicated? What are their Costs: *J. Neurol.* 246, 73–78 (1999)
17. Temple, E.: Brain mechanisms in normal and dyslexic readers. *Current Opinion in Neurobiology* 12, 178–183 (2002)
18. Voeller, K.K.S.: Attention-Deficit / Hyperactivity: Neurobiological and clinical aspects of attention and disorders of attention. In: Coffey, Brumbak (eds.) *Text Book of Pediatric Neuropsychiatry*, pp. 691–718. American Psychiatric Press, Washington (1998)
19. Kwak, Y.T.: Quantitative EEG findings in different stages of Alzheimer's disease. *J. Clin. Neurophysiol.* 23(5), 456–461 (2006)

Features Selection in Character Recognition with Random Forest Classifier

Wladyslaw Homenda^{1,2} and Wojciech Lesinski²

¹Faculty of Mathematics and Information Science, Warsaw University of Technology
Plac Politechniki 1, 00-660 Warsaw, Poland

²Faculty of Mathematics and Computer Science, University of Bialystok
ul. Sosnowa 64, 15-887 Bialystok, Poland

Abstract. Proper image recognition depends on many factors. Features' selection and classifiers are most important ones. In this paper we discuss a number of features and several classifiers. The study is focused on how features' selection affects classifier efficiency with special attention given to random forests. Different construction methods of decision trees are considered. Others classifiers (k nearest neighbors, decision trees and classifier with Mahalanobis distance) were used for efficiency comparison. Lower case letters from Latin alphabet are used in empirical tests of recognition efficiency.

Keywords: pattern recognition, features selection, classification, random forest.

1 Introduction

Image recognition is one of the most important branches of artificial intelligence. Image recognition and related topics had already been researched for decades. Even though in many domains gained results were satisfying, still there are loads of fields to be explored. Proper features' selections could be one of the elements of that problem.

Work of every classifier is based on appropriate description of given object. In case of image classification features' vectors usually describe objects subjected to recognition. Many publications show different features extracted from an image. Those are histograms, transitions, margins, moments, directions and many more. It may seem that creating a vector of all known features would be the best solution. Unfortunately such a vector would cause that huge loads of computation might be necessary and would result in unacceptable time-consuming. On the other hand, often adding next features doesn't increase classifier's efficiency. For that reason, a study no influence of particular features on classifier efficiency is an important element of image recognition.

The paper is aimed on studying how features selection affects efficiency of random forest classifier. The discussion is validated on a basis of thousands of samples of lowercase letters of Latin alphabet. Recognition efficiency of random

forests classifier on this base of samples is faced up to k nearest neighbor, decision tree and Mahalanobis distance classifiers.

The paper is organized as follows. In Section 2 basic concepts concerning decision trees and random forests are outlined. Features are explained in Section 3. Section 4 describes empirical tests.

2 Preliminaries

2.1 Decision Trees

A decision tree is a decision support tool that uses a tree-like graph or model of decision making and classification. Popular algorithms used for construction of decision trees have inductive nature using use top-bottom tree building scheme. In this scheme, building a tree starts from the root of the tree. Then, a feature for testing is chosen for this node and training set is divided to subsets according to values of this feature. For each value there is a corresponding branch leading to a subtree, which should be create on the basis of the proper testing subset. This process stops when a stop criterion is fulfilled and current subtree becomes a leaf. An example algorithm of tree construction is described in the next section.

Stop criterion shows when construction process needs to be brought to a standstill, that is when for some set of samples we should make a leaf, not a node. An obvious stop criterion could be situation when:

- a sample set is empty,
- all samples are from the same class,
- attributes set is empty.

In practice criterions given above sometimes make overfitting to learning data. So then another stop criterions or mechanisms, such as pruning, is necessary to be applied in order to avoid the overfitting problem.

Finally, classification of a given object is based on finding a path from the root to a leaf along branches of the tree. Choices of branches are done by assigning tests' results of the features corresponding to nodes. The leaf ending the path gives the class label for the object, c.f. 3 and 5.

2.2 ID3 Algorithm

ID3 (Iterative Dichotomiser 3) is an algorithm used to generate a decision tree. The algorithm was invented by Ross Quinlan 7. This algorithm uses entropy as a test to divide training set. Entropy for a given set X split to classes $C_1, C_2 \dots C_M$ is as follows:

$$entropy(X) = - \sum_{i=1}^M p_i (\log(p_i)) \quad (1)$$

where $P = (p_1 \dots p_M)$ are probabilities for classes $C_1, C_2 \dots C_M$.

Average entropy in a given node v and for an attribute A_l is defined as follows:

$$avg_entropy(X_v) = \sum_{i=1}^{k_l} \frac{|T_i|}{|X_v|} * entropy(T_i) \quad (2)$$

where $(T_1, T_2, \dots, T_{k_l})$ is a division of the training subset X_v corresponding to the node v attribute A_l , T_i includes testing elements of the subset X_v , which have the value a_{li} of the attribute A_l , and k_l is the number of values of the attribute A_l .

The algorithm ID3 is based on information entropy and can be formulated as follows:

1. put the testing set in the root of the decision tree,
2. if for a given node of the tree all samples belong to the same class C_i , then the node becomes the leaf labelled by the class C_i ,
3. if for a given node the attribute set is empty, then the node becomes the leaf labelled by the class C_i having majority in the testing subset in this node,
4. if for a given node the attribute set is not empty and samples in the testing set are not in same class, then:
 - compute average entropy for each attribute,
 - choose an attribute with minimal entropy,
 - split the testing subset according to values of the chosen attribute,
 - for every set of the split: create the successor of the node and put the set in this node,
 - apply points 2, 3 and 4 for newly created nodes.

2.3 Random Forest

Random forest is a relatively new classifier proposed by Breiman in [2]. The method combines Breiman's [1] "bagging" idea and the random selection of features in order to construct a collection of decision trees with controlled variation.

Random forest is composed of some number of decision trees. Each tree is built as follow:

- let the number of training objects be N , and the number of features in features vector be M ,
- training set for each tree is built by choosing N times with replacement from all N available training objects,
- number $m \ll M$ is an amount of features on which to base the decision at that node. This features is randomly chosen for each node,
- Each tree is built to the largest extent possible. There is no pruning.

Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

3 Features

In this paper we discuss 21 different features of monochrome images. The features are gathered in the following groups:

- projections, 8 features,
- transitions, 2 features,
- margins, 4 features,
- moments, 6 features,
- quarter, 1 feature.

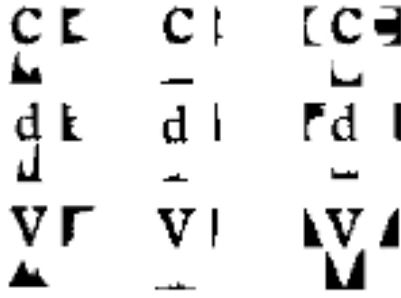


Fig. 1. Letters "c","v","d" and theirs features: horizontal and vertical projections (the left part of the image), horizontal and vertical transitions (the middle part) and left, right and bottom margins(the right part).

Projections. Horizontal and vertical projections in a rectangle are taken. Horizontal projection is defined for every row of pixels of the rectangle. For a given row, the value of the projection is equal to number of black pixels in this row. By analogy, vertical projection is defined in columns of the rectangle, c.f. Figure 1. For both projections, the maximum value, position of the maximum value, the average value and the support (number of nonzero values) are included in the feature's vector.

Transitions. Like in case of projections, horizontal and vertical transitions in a rectangle are taken. Horizontal transitions are defined for every row of pixels of the rectangle. For a given row, the value of the transition is equal to number of pairs of consecutive white and black pixels in this row. By analogy, vertical transitions are defined in columns of the rectangle, c.f. Figure 1. Transitions reflect shape complexity of the image. For both projections, the maximum value, position of the maximum value, the average value and the support (number of nonzero values) are included in the feature's vector. Only maximal values of transitions in both horizontal and vertical directions are included in the features' vector.

Margins. Left, top, right and bottom margins in a rectangle are taken. Left margin is defined for every row of pixels of the rectangle. For a given row, the value of the left margin is equal to the number of white pixels from the left edge of the rectangle right to the first black one. The value of the right margin is equal to the number of white pixels from the right edge of the rectangle left to the first black one. Top and bottom margins are defined analogously, c.f. Figure 1. These features show the symbol's position in the image. We used maximum value of all margins in features vector. Maximum values of all margins are included in the features' vector.

Moments. Moments are used in different fields, e.g. in physics (e.g. mass, center of mass, moment of inertia), in probability (e.g. mean value, variance). In image processing, computer vision and related fields, moment are certain particular

weighted averages of the image pixels' intensities. Also, functions of moments are often utilized in order to have some attractive property or interpretation. Image moments are useful to describe objects after segmentation. Simple properties of the image which are found via image moments including area (or total intensity), its centroid and information about its orientation. The formulas given below define moments of the zero, first and second order. These moments are included in the features' vector. Formula 9 defines a function of moments defining eccentricity of the image, c.f. 6.

Let us assume that for monochrome images for the pixel in row i and column j the mapping I is defined by the formula:

$$I(x_i, y_j) = \begin{cases} 1 & \text{if pixel belongs to the object} \\ 0 & \text{for other points} \end{cases} \quad (3)$$

In other words, for monochrome images the mapping I takes value 1 for black pixels and value 0 for white pixels. Then, the following formulas describe moments included in features' vector:

$$\mu_{00} = \sum_{i=1}^M \sum_{j=1}^N I(x_i, y_j) \quad (4)$$

$$\bar{x} = \mu_{10} = \frac{1}{\mu_{00}} \sum_{i=1}^M \sum_{j=1}^N x_i I(x_i, y_j) \quad (5)$$

$$\bar{y} = \mu_{01} = \frac{1}{\mu_{00}} \sum_{i=1}^M \sum_{j=1}^N y_j I(x_i, y_j) \quad (6)$$

$$\mu_{20} = \frac{1}{\mu_{00}} \sum_{i=1}^M \sum_{j=1}^N (x_i - \bar{x})^2 I(x_i, y_j) \quad (7)$$

$$\mu_{02} = \frac{1}{\mu_{00}} \sum_{i=1}^M \sum_{j=1}^N (y_j - \bar{y})^2 I(x_i, y_j) \quad (8)$$

$$R_c = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}{S} \quad (9)$$

where S is a field of recognized object and μ_{11} is given by the equation 10

$$\mu_{11} = \frac{1}{\mu_{00}} \sum_{i=1}^M \sum_{j=1}^N (x_i - \bar{x})(y_j - \bar{y}) I(x_i, y_j) \quad (10)$$

Quarter. Quarter is the number of image quarter, which includes most of black pixels.

4 Experiment

4.1 Learning and Testing Sets

Experimental results are based on the basis of lowercase Latin letters. The learning set included 1252 images in total. These images represent 26 classes corresponding to all lowercase characters from the Latin alphabet. Every class had different number of elements. The biggest class includes 102 images of "a". The smallest one has only 5 images of "q". Chosen training images are shown in Figure 2.

The testing set includes 2550 images in total. They are dispersed in 26 classes of lowercase Latin letters. Like in the case of the learning set, classes have different numbers of images.

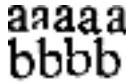


Fig. 2. Chosen representative of training set

4.2 Preprocessing

Symbols undergo preprocessing before classification. Images in gray-scale are converted to monochromatic ones. Conversion is performed with use of a defined threshold. Pixels having value below that threshold obtain value 0 and become white, while pixels with value above that threshold become black and obtain value 1. In the next stage of normalization, an input rectangular image is scaled to the 32×32 square image. The scaling process consists of two stages. In the first stage rectangular image is scaled with aspect ratio maintained, in order to shrink or expand its' longer side to length 32. In the second stage white pixels are added to the shorter side on both sides of the picture, in order to put the center of gravity in the center of newly obtained pattern, c.f. [4].

4.3 Features Independence

Features independence is an important aspect in pattern recognition. Using dependent features in image recognition usually raises such problems as recognition mistakes or unnecessary computational overload. To eliminate depended features we may calculate a correlation matrix, i.e. a matrix of correlation coefficients for all pairs of features.

Elements of a correlation matrix are defined using Pearson product-moment correlation coefficient estimated by formula [11]

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (11)$$

where x_i and y_i are features' values of objects from training set and n is the number of objects.

4.4 Other Classifiers

Empirical results employed in finding the best features' vector should be grounded on more than only one type of classifiers, in our case - on random forests. In this study random forests are 'faced up to other classifiers: k-nearest-neighbors, minimal Mahalanobis distance and decision tree.

k-nearest neighbors. The k-nearest neighbors algorithm is amongst the simplest one of all machine learning algorithms. For a given object being classified its k nearest neighbors of the learning set is encountered. The object is classified to the class having majority in the set of its k nearest neighbors. k is a positive integer, typically small. If k is equal to 1, then the object is simply assigned to the class of its nearest neighbor. Usually this classifier has high recognition rate, but it is run time consuming. Its big computation overload is a significant disadvantage.

Mahalanobis minimal distance. The Mahalanobis minimal distance classifier can be interpreted as a modification of the naive Bayes algorithm. In this method we assume, that a priori likelihood for every class is equal each to other, i.e. $\pi_1 = \pi_2 = \dots = \pi_M$, and all observations come from normal distribution with the same covariance matrixes. With this assumption the Bayes rule assumes the following form:

$$(x - m_k)^T \Sigma^{-1} (x - m_k) \quad m_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_{ik} \quad (12)$$

where x is classified object, m_k is a mean of C_k class calculated from training set and Σ is a covariance matrix defined by equation [13](#)

$$\Sigma = \frac{1}{n - M} \sum_{k=1}^M \sum_{i=1}^{n_k} (x_{ki} - m_k)(x_{ki} - m_k)^T \quad (13)$$

where $x_{1k} \dots x_{n_k}$ are vectors represented objects from class C_k , m_k is a mean vector from this class, n_k - number of elements in class C_k , n - number of all elements and M number of classes.

An equation [12](#) is called Mahalanobis distance. In this method an object x is classified to the class j if square of Mahalanobis distance for this class is the smallest one.

Decision trees. Decision tree has been discussed in Section [2](#).

5 Results

5.1 Character Recognition

Tests were performed for 7 groups of features:

- all 22 features (in Table 1 denoted as "All features"),
- 8 projections based features (Projections),

- 6 margins and transitions based features (Mar-Tran),
- 7 moments based features and quarter (Moments),
- 14 projections, margins and transitions based features (Proj-Mar),
- 15 projections and moments based features (Proj-Mom),
- 13 margins, transitions and moments based features (Mar-Mom),

All object from testing set were recognized by 5 classifiers for every group of features. Characters were classified by:

- random forest with 25 trees,
- random forest with 100 trees,
- k nearest neighbors ($k = 10$ and euclidian distance),
- decision tree,
- minimal Mahalanobis distance.

All results are presented in Table [1](#)

Table 1. Recognition rate for different classifiers and different sets of features

	Forest 100	Forest 25	Mahalanobis	k NN	Tree
Projections	78%	75%	73%	70%	72%
Mar-Tran	81%	78%	60%	75%	71%
Moments	88%	85%	77%	80%	79%
Proj-Mar	88%	85%	84%	84%	81%
Proj-Mom	92%	89%	88%	89%	87%
Mar-Mom	90%	88%	87%	88%	86%
All features	95%	93%	93%	92%	90%

Vector with all available features was tested first of all. In this case best result was obtained by random forest with 100 trees and it was 95%. Random forest with 25 trees and minimal Mahalanobis distance are a little worse reaching 93%. The kNN algorithm is a bit worse reaching 92%. The worst result at the level of 90% is produced by the decision tree.

The next features' vector includes projections based features, i.e. the maximum value, the position of the maximum value and the average value of both vertical and horizontal projections. For this 6 features the best best recognition rate, equal to 75%, is obtained by random forest with 100 trees. The worst recognition rate, equal to 68%, is reached by kNN classifier. Minimal Mahalanobis distance algorithm reached 71% level of recognition. The same result is produced by decision tree. Random forest with 25 trees is a bit better reaching 72%.

The subsequent features' vector includes maximum values of all margins and horizontal and vertical projections. As well as in cases described above, random forest with 100 trees produces the best result with recognition rate equal to 81%. Random Forest with 25 trees is slightly worse at the level of 78%. kNN classifier

achieved 75%, decision tree achieved 71%. The worst result at the level of 60% is obtained by minimal Mahalanobis distance classifier.

The last small group of features includes moments, eccentricity and quarter. As above, the best efficiency is attained by random forest with 100 trees, the recognition rate is equal to 87%. Again, random forest with 25 trees is slightly worse with 84% efficiency. KNN algorithm accomplishes 80% efficiency, decision tree - 79%, classifier with Mahalanobis distance - 77%.

Next groups of features includes 12 elements taken from projections, margins and transitions. Consequently, random forest with 100 trees obtained 86% efficiency while random forest with 25 trees gained 83%. Minimal Mahalanobis distance and kNN classifiers attained 83% efficiency. In this case decision tree is the worst classifier with 79% efficiency.

Another group of features is based on projections and moments. Also, in this case, random forest with 100 trees had the best efficiency at the level of 91%. kNN classifier get 89% recognition rate. Minimal Mahalanobis distance classifier is a bit worse with 88% rate. The worst results, 85%, is reached by decision tree.

Last features' group is built on margins, transitions and moments. The highest efficiency is reached by random forest with 100 trees and it is 90%. Decision tree had the lowest result equal to 85%.

5.2 Correlation Matrix

Correlation matrix is calculated in order to find dependent features. This matrix includes Pearson's product-moment correlation coefficients for each pair of variables. Coefficient is calculated by SPSS 14.0pl for Windows [8]. It occurs that maximal value of correlation coefficients (about 0.5) is between maximal values of horizontal and vertical projections. Pairs built by projections' width and margins also had high correlation coefficient, which is about 0.4. This high values of correlation coefficient may be raised by characters shape or normalization process. Other pairs of features have values of correlation coefficient less than 0.4 (most of them are less than 0.2). In conclusion we can say, that our features are weakly dependent.

6 Conclusions

The study on influence of features selection on effectiveness of different classifiers is presented in this work. Results of empirical tests show that random forest classifier gains the best result comparing to tested methods. For each group of features its efficiency is highest.

It occurs that classifiers strongly depends on the group of features, which includes moments. The group consisting of margins and transitions is less dependent than the one with moments. The group of features consisting of projections gives the worst results. Tests show that the group of all features gives the best results. Recognition rate is significantly higher for all features involved in classification. This leads to the conclusion that the set of features tested is not overrepresented. The conclusion is supported by weak correlation between features.

As the final conclusion, we can say that adding more features may improve efficiency. Future directions on classification of images by random forest would be focused on construction of better features vectors.

Acknowledgments. This work is supported by The National Center for Research and Development, Grant no N R02 0019 06/2009.

References

1. Breiman, L.: Bagging predictors. *Machine Learning* 26(2), 123–140 (1996)
2. Breiman, L.: Random Forests. *Machine Learning* 45, 5–32 (2001)
3. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*. John Wiley & Sons Inc., New York (2001)
4. Homenda, W., Lesinski, W.: Optical Music Recognition: Case of Pattern Recognition with Undesirable and Garbage Symbols. In: Choras, R., et al. (eds.) *Image Processing and Communications Challenges*, pp. 120–127. Exit, Warsaw (2009)
5. Koronacki, J., Cwik, J.: *Statystyczne systemy uczace sie*. Exit, Warsaw (2008) (in Polish)
6. Malina, W., Smiatacz, M.: *Metody cyfrowego przetwarzania obrazow*. Exit, Warsaw (2005) (in Polish)
7. Quinlan, J.R.: Induction of Decision Trees. *Machine Learning* 1, 81–106 (1986)
8. <http://www.spss.com>

Generating and Postprocessing of Biclusters from Discrete Value Matrices

Marcin Michalak and Magdalena Stawarz

Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland
{Marcin.Michalak,Magdalena.Stawarz}@polsl.pl

Abstract. This paper presents a new approach for the biclustering problem. For this purpose new notions like half-bicluster and biclustering matrix were developed. Results obtained with the algorithm BicDM (Biclustering of Discrete value Matrix) were compared with some other methods of biclustering. In this article the new algorithm is applied for binary data but there is no limitation to use it for other discrete type data sets. In this paper also two postprocessing steps are defined: generalization and filtering. In the first step biclusters are generalized and after that only those which are the best become the final set - weak biclusters are filtered from the set. The usage of the algorithm makes it possible to improve the description of data with the reduction of bicluster number without the loss of information. The postprocessing was performed on the new algorithm results and compared with other biclustering methods.

Keywords: machine learning, data mining, biclustering, postprocessing.

1 Introduction

Data biclustering may be considered as a twodimensional partial generalization of a clustering problem. In the case of typical onedimensional clustering we have a set of objects and we try to assign to each object the label from the finite set of labels. Objects with the same label are considered to be in the same cluster [10]. In the case of biclustering we have two sets of objects (that will be called features and co-features) that generate the cartesian product of ordered pairs. There is a value assigned to every pair and biclustering means the search of subsets of features and co-features, which cartesian product pairs would have similar (inexact bicluster) or the same (exact bicluster) values [8, 4].

This article describes the new way of biclustering: BicDM (Biclustering of Discrete value Matrix). It is based on notions feature and co-feature that substitute "row" and "column" notions and on the notion of half-bicluster that is the subset of elements from feature (or co-feature) set. As it results from the name of the algorithm it is designed for data with discrete values. We will present its application for binary matrices (matrices that contain only 0 and 1 values).

This paper introduces also the algorithm of biclusters postprocessing, dedicated for BicDM because the part of it needs some data structures created in

the one of BicDM step. The postprocessing is divided into two parts: biclusters generalization (BicDM dependent part) and biclustering filtration that is more general. As the first part tries to combine biclusters the second one tries to remove the less important ones. This step is similar to the well known practice of the decision rules filtration [16].

This paper is organized as follows: in the next section some foundations of biclustering are described with some standard algorithms. Then some new notions that deal with biclustering are introduced. Next part contains the description of the new biclustering algorithm including the problem of using it for matrices with more than two discrete values called non-binary. Then, the both of biclustering postprocessing steps are defined. Afterwards, BicDM results are compared with biclusters generated by other algorithms from binary matrices. Then the short discussion about the application of multi-agent systems for biclustering is written. Finally, some ending conclusions and remarks are described.

2 Biclustering Problem

Biclustering is a data mining method which allows to find similar subset of rows across similar subset of columns that are correlated together [1]. Biclustering algorithms are the most widely used in DNA microarray data analysis [12], text mining [3] and collaborative filtering [14]. In these areas there exists quite a lot of methods that are trying to find the best answer for this problem, but more effective solutions are still highly desirable [9] [18]. Most of them try to solve NP-complete problems, but despite this they require large-scale computational effort or need to employ heuristic methods [9]. To compare our results with existing algorithms we chose two of them called xmotif and OPSM. Both of them are based on greedy search strategies [13].

2.1 Order Preserving Submatrix Algorithm

In this method, bicluster is called as order-preserving submatrix (OPSM) and can be defined as subgroup of objects which across subgroup of attributes can be arranged in such a way that sequence of values in every row rigorously grow up. This algorithm was designed for finding large and statistically significant biclusters among dataset, satisfying strict OPSM requirements. Algorithm allows to determine more than one OPSM based on the same data set, even if they are overlapping. Because problem of finding OPSM is NP-hard, this method applied heuristic approach to solve it [2].

2.2 Xmotif

Authors of this algorithm proposed a new idea for representing data in the form of xmotif [11]. Bicluster called xmotif consists of conserved objects, which values are almost unvarying across subset of attributes. Before algorithm computes xmotif, in the first instance it determines statistically significant extent of values called states, matching to each object. After that to identify valid bicluster, it singles

out conserved objects, states that this objects belong to them and attributes that match the motif. They define the notion of maximal bicluster, which contains the maximal number of conserved objects. After each iteration, attributes which belong to maximal bicluster have been removed and among remaining attributes algorithm looks for the next optimal motif. This procedure goes on until all attributes have not satisfy some motif. Algorithm founds the significant maximal motif based on various random seeds in an repetitive search way.

3 Biclustering Notions

Two-dimensional matrix \mathbb{A} with rows and columns will be denoted as the ordered pair of the set of features \mathcal{F} and the set of co-features \mathcal{F}^* . Rows may be considered as features and columns as co-features but also the opposite assignment is correct. Then, if columns are features, rows are cofeatures. As the bicluster \mathbb{B} of the matrix $\mathbb{A} = \langle \mathcal{F}, \mathcal{F}^* \rangle$ the ordered pair $\langle \mathcal{G}, \mathcal{G}^* \rangle$ is considered that satisfies the following conditions: $\mathcal{G} \subseteq \mathcal{F}$, $\mathcal{G}^* \subseteq \mathcal{F}^*$. We may also claim that bicluster \mathbb{B} is the subcluster of the matrix \mathbb{A} .

3.1 Biclustering Matrices

Let us assume that \mathbb{A} is a matrix with values 0 and 1 (binary matrix): $\mathbb{A}(i, j) \in \{0, 1\}$ $i = 1, 2, \dots, |\mathcal{F}|, j = 1, 2, \dots, |\mathcal{F}^*|$, where $|\cdot|$ is the cardinality of the set. Biclustering matrix may be calculated for features \mathcal{F} or for co-features \mathcal{F}^* . If the matrix has $|\mathcal{F}|$ features and $|\mathcal{F}^*|$ co-features the biclustering matrix for features (*fBM* – feature biclustering matrix) is $|\mathcal{F}| \times |\mathcal{F}^*|$. This matrix is nonsymmetric and is defined as follows: $fBM(i, j) = \{a \in \mathcal{F} : cf_j(a) = 1 \wedge cf_i(a) = 0, cf_i, cf_j \in \mathcal{F}^*\}$ where $a \in \mathcal{F} : cf_j(a) = 1$ means that if a is a feature and cf_j is a co-feature their intersection is equal to 1. When features are rows and co-features are columns it means there is a value 1 in the column j and a row a in the original matrix. In other words it may be said that $fBM(i, j)$ is the set of features which value for co-feature cf_j is 1 and for the co-feature cf_i is 0. Similarly the co-feature biclustering matrix ($cfBM_{|\mathcal{F}^*| \times |\mathcal{F}^*|}$) is defined: $cfBM(i, j) = \{u \in \mathcal{F}^* : f_j(u) = 1 \wedge f_i(u) = 0, f_i, f_j \in \mathcal{F}\}$.

3.2 Biclustering Half-Biclusters

If a square biclustering matrix $BM_{m \times m}$ is given (whether *fBM* or *cfBM*) we may define the half-bicluster for this matrix. As the half-biclustering function for the biclustering matrix we define $h = \bigvee \{ \bigwedge BM(i, j) | BM(i, j) \neq \emptyset \}$ where $\bigwedge BM(i, j)$ represents the logical conjunction of all the items in an element $BM(i, j)$. On the basis of the h function the half-bicluster is defined as follows: for each prime implicant of the function h there exists the half-bicluster such that the conjunction of all attributes forming that half-bicluster is equivalent to the prime implicant.

If the half-bicluster is obtained from the feature biclustering matrix it is called the feature half-bicluster. If the half-bicluster is obtained from the co-feature biclustering matrix it is called the co-feature half-bicluster.

Set of all feature half-biclusters for the matrix \mathbb{A} will be denoted as $f(\mathbb{A})$ and set of all co-feature half-biclusters will be denoted as $f^*(\mathbb{A})$.

3.3 Bicluster Quality Measures

For a given bicluster $\mathbb{B} = \langle \mathcal{G}, \mathcal{G}^* \rangle$ that is the subcluster of the matrix $\mathbb{A} = \langle \mathcal{F}, \mathcal{F}^* \rangle$ two simple quality measures may be defined: accuracy and coverage. Let us define following notions: **matrix weight**: $w(\mathbb{A})$ — number of ones in the matrix \mathbb{A} , **bicluster area**: $\overline{\mathbb{B}} = |F| \cdot |Q|$ — number of ones and zeros in the bicluster \mathbb{B} , **bicluster weight**: $w(\mathbb{B})$ — number of ones in the bicluster \mathbb{B} .

Now we can define bicluster accuracy as the ratio of its weight and area while the bicluster coverage as the ratio of the bicluster weight and the whole matrix weight: $acc(\mathbb{B}) = w(\mathbb{B})/\overline{\mathbb{B}}$; $cov(\mathbb{B}) = w(\mathbb{B})/\overline{\mathbb{A}}$.

4 Biclustering Algorithm

4.1 Biclustering for Binary Matrices

The new biclustering algorithm finds bicluster of ones among the binary twodi-dimensional matrix. It may be divided into three parts. As the initial step some preprocessing of the input information system must be performed. In this step all features (and co-features) with weight (number of ones) equal to zero should be removed from the matrix. This is quite intuitive as the target of the biclustering is to find subclusters of ones. In the same step also features (and co-features) that consist only of ones should be removed. But in this case we keep the information about the removed items because it will be used after the second step of the algorithm. We will call this features (co-features) as core features (co-features).

In the second step of the algorithm feature half-biclusters and co-feature half-biclusters are calculated. Then, if there are any core features each of them is joined to each of the feature half-bicluster. Similarly, if there are any core co-features each of them is included to every co-feature bicluster.

In the final step we merge all feature half-biclusters with every co-feature half-bicluster. This means that the final set of biclusters \mathcal{B} is the cartesian product of the half-biclusters sets: $\mathcal{B}(\mathbb{A}) = f(\mathbb{A}) \times f^*(\mathbb{A})$. This leads to the equivalent definition of the bicluster: $\mathbb{B}(\mathbb{A}) = \langle \mathcal{G}, \mathcal{G}^* \rangle : \mathcal{G} \in f(\mathbb{A}) \wedge \mathcal{G}^* \in f^*(\mathbb{A})$.

It is obvious that $w(\mathbb{B}) = 0$ for some $\mathbb{B} \in \mathcal{B}$. If there are at least two expected biclusters they should generate two feature half-biclusters and two co-feature half-biclusters. This causes that $card(\mathcal{B}) = 4$ but two biclusters do not contain any „one”. From this point of view all „weightless” biclusters should not be considered as proper ones.

4.2 Biclustering for Non-binary Matrices

As it was mentioned before BicDM may also be applied to biclustering non-binary data. If we assume that V is a set of all discrete values in the given matrix then for each $v \in V$ we can determine set of biclusters in $d = |V|$ iterations. For

this purpose we need to introduce step of preprocessing the given matrix that consists of replacing all discrete values from the set $V \setminus \{v\}$ with zero. Afterwards, the algorithm can be used in each iteration to find biclusters among dataset with two discrete values $V_v = \{v, 0\}$. Based on this procedure we can obtain set of biclusters for all discrete values in given matrix separately.

5 Postprocessing of Biclusters

5.1 Biclusters Generalization

Based on two sets of half-biclusters determined by BicDM we can decide which biclusters could be merged in order to obtain more general biclusters which maybe are less accurate but cover the bigger area of the input matrix. This analysis is based on biclustering matrices and half-biclusters for features and co-features. In the first step we will analyse co-feature biclustering matrix. For each $p \in f^*(\mathbb{A})$ we will analyse submatrix of fBM defined as $\forall p_k \in f^*(\mathbb{A}), sub\ cfBM_k = cfBM(:, p)$ where notation of $BM(:, p)$ is similar to the Matlab notation: „:” means all rows or columns of the matrix and p is the subset of rows (columns) determined by the half-bicluster p . Now in each column of this submatrix we are looking for feature half-biclusters $p^* \in f(\mathbb{A})$. If some feature half-bicluster p^* appear in each column in submatrix $sub\ cfBM$, then we can assume that this p^* can be connected with p . Set of p^* founded in $sub\ cfBM$ for one $p \in f^*(\mathbb{A})$ we called SAR_i^* .

When we have results of this analysis for each $p \in f^*(\mathbb{A})$ than we can summarize which $p \in f^*(\mathbb{A})$ could be connected with the same $p^* \in f(\mathbb{A})$. If at least two half-biclusters p have not empty intersection of their SAR then based on this knowledge we can create new biclusters which rows and columns are determined by features and co-features (sum of sets) belonging to p_i, p_j, \dots and SAR_i, SAR_j, \dots

In the second step we should repeat this same procedure as in first the step but on the basis of the fBM . The final result of biclusters generalization is the sum of sets of biclusters obtained in two mentioned steps.

5.2 Biclusters Filtering

It is possible that after the double generalization of the set of biclusters the result set may contain some worse or overlapping biclusters. The second step of biclusters postprocessing consist in filtering the set of bicluster given by the generalization. This algorithm may be compared with typical algorithms of decision rules filtering "from coverage" [15]. It assumes that for each decision class the rules ranking is built that points the best rule that is moved to the result set. After that the ranking is rebuilt as it refers only for object not recognized by rules from the results set. This loop is performed till all objects are covered by result rules.

Biclusters filtering introduces a small modification of the algorithm above. As each bicluster may be described with its accuracy and coverage the ranking of

biclusters will be built on the basis of the WS quality function [17] that takes into consideration both of these values. This quality function is defined as follows: $q_M = (1 - w) \cdot acc + w \cdot cov, w \in [0, 1]$. After each step the „ones” covered by result biclusters are excluded from the evaluation of the quality function.

6 Experiments and Results

Two synthetic data sets were used for the biclustering: Example1 and Example2 (Table 1). For every set and for every biclustering method the number of biclusters, their weight, accuracy and coverage were calculated. Results are presented in Tables 2 and 3. The number in brackets under every method name is the total number of generated biclusters.

Table 1. Synthetic data sets

Example1						Example2															
	P_1	P_2	P_3	P_4	P_5	N_1	N_2	N_3	N_4	N_5		P_1	P_2	P_3	P_4	P_5	N_1	N_2	N_3	N_4	N_5
1	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1
2	0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	0	1	0	1	1	1
3	0	0	0	0	0	1	1	1	1	1	1	1	0	1	0	1	1	0	1	1	0
4	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	0	1	1
5	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1
6	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
7	1	1	1	1	1	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	1
8	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0
9	1	1	1	1	1	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	0
10	1	1	1	1	1	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0

Table 2. Results for Example1

Method	Number of biclusters	area	weight	acc	cov	biclusters
xmotif (2)	1	50	25	0.50	0.51	$\langle U, \{P_1, P_2, P_3, P_4, P_5\} \rangle$
	1	45	15	0.33	0.31	$\langle U \setminus \{2\}, \{N_1, N_2, N_3, N_4, N_5\} \rangle$
OPSM (2)	1	50	25	0.50	0.51	$\langle \{6, 7, 8, 9, 10\}, A \rangle$
	1	40	20	0.50	0.41	$\langle U, \{P_1, P_2, P_3, P_4, P_5\} \rangle$
BicDM (4)	1	25	25	1.00	0.51	$\langle \{6, 7, 8, 9, 10\}, \{P_1, P_2, P_3, P_4, P_5\} \rangle$
	2	4	4	1.00	0.08	$\langle \{1, 3, 4, 5\}, \{N_3\} \rangle, \langle \{2\}, \{N_1, N_2, N_4, N_5\} \rangle$
	1	16	16	1.00	0.33	$\langle \{1, 3, 4, 5\}, \{N_1, N_2, N_4, N_5\} \rangle$

We may see that for the binary matrix with the high level of ones aggregation all three methods of biclustering gave quite small number of biclusters. It is worth to notice that only BicDM algorithm gave the most accurate biclusters. Even if we consider only two BicDM biclusters with the highest weight it occurs that they cover the same part of ones as other methods biclusters without any

Table 3. Results for Example2

Method	Number of biclusters	area	weight	acc	cov	biclusters
xmotif (8)	1	20	10	0,50	0,20	$\langle\{1, 5, 6, 10\}, \{N_1, N_2, N_3, N_4, N_5\}\rangle$
	1	15	10	0,33	0,10	$\langle\{1, 5, 6\}, \{P_1, P_2, P_3, P_4, P_5\}\rangle$
	2	10	5	0,50	0,10	$\langle\{2, 3\}, \{P_1, P_3, P_5, N_1, N_3\}\rangle$ $\langle\{2, 7\}, \{P_1, P_2, P_3, P_5, N_1\}\rangle$
	2	6	0	1,00	0,12	$\langle\{3\}, \{P_1, P_3, P_5, N_1, N_3, N_4\}\rangle$ $\langle\{7\}, \{P_1, P_2, P_3, P_5, N_1, N_5\}\rangle$
	2	5	0	1,00	0,10	$\langle\{4\}, \{P_5, N_1, N_2, N_4, N_5\}\rangle$ $\langle\{9\}, \{P_1, P_2, P_3, N_1, N_4\}\rangle$
OPSM (8)	1	20	10	0,50	0,20	$\langle\{1, 5\}, A\rangle$
	1	27	14	0,48	0,27	$\langle\{1, 4, 5\}, A \setminus N_3\rangle$
	1	32	16	0,50	0,33	$\langle\{1, 2, 4, 5\}, \{A \setminus \{P_4, N_3\}\}\rangle$
	1	30	16	0,47	0,29	$\langle\{1, 2, 4, 5, 7\}, \{P_1, P_2, P_3, P_5, N_1, N_5\}\rangle$
	1	30	15	0,50	0,31	$\langle\{1, 2, 4, 5, 8, 10\}, \{P_2, N_1, N_2, N_4, N_5\}\rangle$
	1	28	17	0,39	0,22	$\langle\{1, 2, 4, 5, 6, 7, 8\}, \{P_1, P_2, P_3, P_5\}\rangle$
	1	27	12	0,56	0,31	$\langle U, \{P_1, P_3, P_5\}\rangle$
BicDM (33)	13	1	1	1,00	0,02	$\langle\{6\}, \{P_2\}\rangle \langle\{9\}, \{P_2\}\rangle \langle\{2\}, \{P_4\}\rangle \langle\{6\}, \{P_4\}\rangle$ $\langle\{8\}, \{P_4\}\rangle \langle\{6\}, \{P_5\}\rangle \langle\{7\}, \{P_5\}\rangle \langle\{8\}, \{P_5\}\rangle$ $\langle\{9\}, \{P_5\}\rangle \langle\{8\}, \{N_3\}\rangle \langle\{2\}, \{N_4\}\rangle$ $\langle\{9\}, \{N_4\}\rangle \langle\{7\}, \{P_2\}\rangle$
	10	2	1	0,50	0,02	$\langle\{N_5, P_3\}, \{8\}\rangle \langle\{N_3\}, \{10, 3\}\rangle \langle\{P_4\}, \{10, 3\}\rangle$ $\langle\{N_1, N_5\}, \{10, 3\}\rangle \langle\{N_4\}, \{10, 3\}\rangle \langle\{N_1, N_5\}, \{9\}\rangle$ $\langle\{P_3, N_5\}, \{2\}\rangle \langle\{N_2, N_5\}, \{7\}\rangle$ $\langle\{N_1, N_5\}, \{2\}\rangle \langle\{N_5, P_3\}, \{6\}\rangle$
	6	2	2	1,00	0,04	$\langle\{7\}, \{P_3, N_5\}\rangle \langle\{3, 10\}, \{P_5\}\rangle \langle\{7\}, \{N_1, N_5\}\rangle$ $\langle\{2\}, \{N_2, N_5\}\rangle \langle\{1, 5\}, \{N_3\}\rangle \langle\{1, 5\}, \{N_4\}\rangle$
	2	4	2	0,50	0,04	$\langle\{10, 3\}, \{N_5, P_3\}\rangle \langle\{1, 5\}, \{N_5, P_3\}\rangle$
	2	4	4	1,00	0,08	$\langle\{1, 5\}, \{N_1, N_5\}\rangle \langle\{1, 5\}, \{N_2, N_5\}\rangle$

Table 4. Postprocessing results for Example1

Method	Bicl. No.	area	weight	acc	cov	bicluster
xmotif (2)	#1	50	25	0.5	0.51	$\langle U, \{P_1, P_2, P_3, P_4, P_5\}\rangle$
	#1	45	20	0.44	0.41	$\langle\{U \setminus \{2\}\}, \{N_1, N_2, N_3, N_4, N_5\}\rangle$
OPSM (2)	#1	50	25	0.5	0.51	$\langle\{6, 7, 8, 9, 10\}, A\rangle$
	#2	40	20	0.5	0.41	$\langle U, \{P_1, P_2, P_3, P_5\}\rangle$
RB (2)	#1	25	25	1.00	0.51	$\langle\{6, 7, 8, 9, 10\}, \{P_1, P_2, P_3, P_4, P_5\}\rangle$
	#2	25	24	0.96	0.49	$\langle\{1, 2, 3, 4, 5\}, \{N_1, N_2, N_3, N_4, N_5\}\rangle$

decrease of accuracy. From the other hand if ones are sparse arranged in the data set BicDM algorithm still returns accurate biclusters but with very small coverage.

After the BicDM results generalization we obtained 3 biclusters for the Example1 and 19 for the Example2. These sets were the input for the filtration part of the postprocessing. After this step the bicluster description of the Example1 set contained two biclusters and the description of the Example2 contained four biclusters. More detailed information about the filtering result is presented in Tables 4 and 5.

We may see that the number of BicDM biclusters decreased significantly. For the Example1 we have two important advantages: the number of biclusters is equal to the number of biclusters generated by the other two methods and post-processed biclusters have better quality (both accuracy and coverage). Similar

Table 5. Postprocessing results for Example2

Method	Bicl. No.	area	weight	acc	cov	bicluster
xmotif (6)	#1	6	6	1.00	0.12	$\langle\{3\}, \{P_1, P_3, P_5, N_1, N_3, N_4\}\rangle$
	#2	6	6	1.00	0.12	$\langle\{7\}, \{P_1, P_2, P_3, P_5, N_1, N_5\}\rangle$
	#3	5	5	1.00	0.10	$\langle\{4\}, \{P_5, N_1, N_2, N_4, N_5\}\rangle$
	#4	5	5	1.00	0.10	$\langle\{9\}, \{P_1, P_2, P_5, N_1, N_4\}\rangle$
	#5	20	10	0.50	0.20	$\langle\{1, 5, 6, 10\}, \{N_1, N_2, N_3, N_4, N_5\}\rangle$
	#6	15	5	0.33	0.10	$\langle\{1, 5, 6\}, \{P_1, P_2, P_3, P_4, P_5\}\rangle$
OPSM (6)	#1	27	15	0.56	0.31	$\langle U, \{P_1, P_3, P_5\}\rangle$
	#2	30	15	0.50	0.31	$\langle\{1, 2, 4, 5, 8, 10\}, \{P_2, N_1, N_2, N_4, N_5\}\rangle$
	#3	28	9	0.32	0.18	$\langle\{1, 2, 4, 5, 6, 7, 8\}, \{P_1, P_2, P_3, P_5\}\rangle$
	#4	20	3	0.15	0.06	$\langle U, \{P_1, P_5\}\rangle$
	#5	20	2	0.10	0.04	$\langle\{1, 5\}, A\rangle$
	#6	30	2	0.07	0.04	$\langle\{1, 2, 4, 5, 7\}, \{P_1, P_2, P_3, P_5, N_1, N_5\}\rangle$
RB (4)	#1	6	6	1.00	0.12	$\langle\{3, 6, 7, 8, 9, 10\}, \{P_5\}\rangle$
	#2	4	4	1.00	0.08	$\langle\{7\}, \{P_2, P_3, N_1, N_5\}\rangle$
	#3	18	14	0.78	0.29	$\langle\{1, 2, 5\}, \{P_4, N_1, N_2, N_3, N_4, N_5\}\rangle$
	#4	48	15	0.31	0.31	$\langle\{3, 6, 7, 8, 9, 10\}, A \setminus \{P_1, N_2\}\rangle$

situation takes the place for the Example2: the number of biclusters was reduced almost ten times and now it is two times lower then the number of biclusters generated by xmotif and OPSM. In both cases postprocessed biclusters cover the same „ones” as not postprocessed biclusters.

7 CCI Perspectives of Biclustering

If we assume that matrix $\mathbb{A} = \langle \mathcal{F}, \mathcal{F}^* \rangle$ contains $d = |V|$ discrete values the computational complexity of biclustering algorithm (excluding the postprocessing step) is $O(d|\mathcal{F}||\mathcal{F}^*|(|\mathcal{F}| + |\mathcal{F}^*|))$. This fact implies that the natural way of the algorithm improvement is to divide the problem into d subtasks and each of them should be calculated separately and concurrently. The more advanced model of biclustering decomposition assumes that the input matrix is divided into several disjoint matrices and each agent finds biclusters in the pointed submatrix. Then the result biclusters are joined in the other agent.

It is not said that the new algorithm always generates the best possible biclusters. It means that it is still worth to consider results obtained with other methods. A very simple architecture of multi-agent system may contain one agent for every biclustering method and one agent for postprocessing of all results. This structure may be also extended with the set of agents that postprocess biclusters from the only one method. Only partially postprocessed biclusters are sent to the common agent that combines results from different methods.

8 Conclusions and Further Works

In this article the new algorithm of biclustering (BicDM) was presented. It is designed for biclustering data from binary twodimensional matrices. BicDM represents the inexact approach for the problem of data biclustering. This new algorithm is designed for finding biclusters of ones among the twodimensional binary

matrices. For other types of analysis there exist some simple modifications that reduce the task to BicDM conditions. Biclusters of zeros may be found as biclusters of ones in the negation of input matrix. Biclustering of non-binary should be performed for every discrete value separately. Biclustering of numeric (whether continuous or not) should be preceded with some discretization process.

The definition of half-biclusters assures that none of the bicluster is the subset of the other one. The same definition allows two biclusters to have the non empty intersection.

We may see that for data sets with quite connected biclusters BicDM gives very good results: small number of big and disjoint biclusters. These four biclusters cover the whole set of ones and every pair of them is disjoint. None of compared algorithms gave comparable results. From the other side when ones are distributed sparse BicDM generates a big number of clusters. The solution of this problem is the postprocessing of obtained biclusters analogically as it takes place in the case of decision rules postprocessing. In this step small accurate rules are joined to the more general one with the higher coverage with the permit of a small accuracy decrease.

Results of biclusters postprocessing look very satisfactory: the number of biclusters was significantly reduced without the loss of the description accuracy and the data coverage. It is obvious that the result set may vary as the quality function that generates the bicluster ranking has one free and modifiable parameter. It is user intention to prefer biclusters that are more accurate or have the bigger coverage.

Our further works will focus on more advanced bicluster filtering methods and on the generalization algorithm that will be applicable also for biclusters obtained with the usage of other algorithm. We are also working on the problem of joining biclusters that are generated by agents from disjoint submatrices.

We also plan to generalize the BicDM for matrices of values from discrete but ordered set and for matrices with real values. First problem seems to be similar to the well known problem of dominance [7] or quasi-dominance [6] models of rough sets theory. The second one may be based on other similar models for rough sets analysis like the tolerance model [16] or variable precision model [5].

Acknowledgments. This work was supported by the European Community from the European Social Fund. Special thanks also to Marek Sikora and Krzysztof Cyran for some rough sets theory advice.

References

1. Ayadi, W., Elloumi, M., Hao, J.K.: A biclustering algorithm based on a Bicluster Enumeration Tree: application to DNA microarray data. *BioData Mining* 9 (2009)
2. 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)
3. Chang, F.C., Huang, H.C.: A refactoring method for cache-efficient swarm intelligence algorithms. *Inf. Sci.* (2010), doi:10.1016/j.ins.2010.02.025 (in press, corrected proof)

4. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proc. of the 8th Int. Conf. On Intell. Systems For Molecular Biology, pp. 93–103 (2000)
5. Cyran, K.: Modified Indiscernibility Relation in the Theory of Rough Sets with Real-Valued Attributes: Application to Recognition of Fraunhofer Diffraction Patterns. In: Peters, J.F., Skowron, A., Rybiński, H. (eds.) Transactions on Rough Sets IX. LNCS, vol. 5390, pp. 14–34. Springer, Heidelberg (2008)
6. Cyran, K.: Quasi Dominance Rough Set Approach in Testing for Traces of Natural Selection at Molecular Level. *Advanc. in Intell. and Soft. Comput.* 59, 163–172 (2009)
7. Greco, S., Matarazzo, B., Slowinski, R.: Rough approximation of a preference relation by dominance relations. *Eur. J. of Oper. Res.* 117, 63–83 (1999)
8. Hartigan, J.A.: Direct clustering of a data matrix. *J. of the Am. Stat. Assoc.* 67, 123–129 (1972)
9. Madeira, S.C., Oliveira, A.L.: Biclustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Trans. on Comput. Biol. and Bioinforma.* 1, 24–45 (2004)
10. Mirkin, B.: Mathematical Classification and Clustering. *J. of the Oper. Res.* 48, 852–853 (1997)
11. Murali, T.M., Kasif, S.: Extracting Conserved Gene Expression Motifs from Gene Expression Data. In: Pacific Symposium on Biocomputing, pp. 77–88 (2003)
12. Nisar, A., Ahmad, W., Liao, W., Choudhary, A.: High Performance Parallel/Distributed Biclustering Using Barycenter Heuristic. In: Proc. of SIAM Int. Conf. on Data Mining, pp. 1050–1061 (2009)
13. 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, 1122–1129 (2006)
14. Sayoud, H., Ouamour, S.: Speaker Clustering of Stereo Audio Documents Based on Sequential Gathering Process. *J. of Inf. Hiding. and Multimed. Signal Process.* 4, 344–360 (2010)
15. Sikora, M.: An algorithm for generalization of decision rules by joining. *Found. on Comput. and Decis. Sci.* 30, 227–239 (2005)
16. Sikora, M.: Decision Rule-Based Data Models Using TRS and NetTRS – Methods and Algorithms. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets XI. LNCS, vol. 5946, pp. 130–160. Springer, Heidelberg (2010)
17. Sikora, M.: Filtering of decision rules using rules quality function. *Stud. Inform.* 4(46), 5–21 (2001)
18. Tanay, A., Sharan, R., Shamir, R.: Biclustering algorithms: A survey. In: Handbook of computational molecular biology. Chapman Hall/CRC Press (2006)

A Validity Criterion for Fuzzy Clustering

Stanisław Brodowski

Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian
University Krakow, Poland
stanislaw.brodowski@uj.edu.pl

Abstract. This paper describes a new validity index for fuzzy clustering: *Pattern Distances Ratio (PDR)* and some modifications improving its performance as cluster number selection criterion for Fuzzy C-means. It also presents experimental results concerning them.

As other validity indices, solution presented in this paper may be used when a need for assessing of clustering or fuzzy clustering result adequacy arises. Most common example of such situation is when clustering algorithm that requires certain parameter, for example number of clusters, is selected but we lack a priori knowledge of this parameter and we would use educated guesses in concert with trial and error procedures. Validity index may allow to automate such process whenever it is necessary or convenient. In particular, it might ease incorporation of fuzzy clustering into more complex, intelligent systems.

Keywords: clustering, fuzzy, validity index, number of clusters.

1 Introduction

A general goal of clustering is to divide a set of patterns into such groups that examples within one group are similar one to another, while being relatively dissimilar to examples from other groups. The search for such structure is basically unsupervised and without help of some special (e.g. response) variables.

Fuzzy clustering uses fuzzy sets to represent clusters so, instead of just assigning each pattern to a cluster, it assigns each pattern some membership value to each cluster. For i -th example in a training series D and c clusters, μ_i is the membership vector: $\mu_i = (\mu_{i1} \dots \mu_{ic})^T$.

Usually $\mu_{ij} \in [0, 1]$ for each cluster and pattern. Commonly used *probabilistic cluster partition* [1] demands also that: $\sum_{j=1}^c \mu_{ij} = 1, \quad \forall i : x_i \in D$.

To proceed with clustering one needs to at least specify what words “similar” and “dissimilar” mean in the definition mentioned at the beginning of this section. This usually includes selecting a distance measure (Euclidean or Manhattan metric, Mahalanobis distance etc.) and some means to assess what is “similar enough” (e.g. linkage type or some exact function to be minimized by the algorithm). Still, some algorithms require more prior information. For example, well known K-means [1] and Fuzzy C-means [2,3,1] require the number of clusters to divide the data into.

Direct computation or estimation of desired parameters' values may be impractical because of time or space complexity or difficulties in implementation. In such case, a technique to at least assess a given set of parameters' values could be of great help. *Validity indices* attempt to do that by analyzing results of the clustering algorithm applied with parameters set to given values [1]. It means that actual clustering has to take place for the values to be assessed, but also makes automation of the search for best values possible.

This is especially important if the clustering is called multiple times as a part of a more complex solution that should not be supervised for some reason. An entity in collective intelligence system like intelligent software agent in a multi-agent system could be a good example, as including excessive external non-automatic supervision in any form into such system may drastically decrease its usefulness. Some hybrid systems (like [4]) could also benefit from validity index use. Validity indices can also have other uses – e.g. to estimate quality of clustering algorithm.

This paper presents a fuzzy clustering validity index called Pattern Distances Ratio (*PDR*) and evaluates its performance as cluster number selection criterion for Fuzzy C-means on several datasets. In Sect. 2 it describes the general function *PDR* and certain improvements for a special case of using it as cluster number selection criterion for Fuzzy C-means. As experiments in Sect. 3 show, this index appears to be sensitive for certain clustering tendencies that other tested criteria do not detect and performs well on standard datasets.

Fuzzy C-means. Fuzzy C-means attempts to minimize, in iterative, alternating manner, the objective function [1]:

$$J_m = \sum_{j=1}^c \sum_{i=1}^{|D|} \mu_{ij}^m d(x_i, c_j),$$

where d is a distance measure (often Euclidean distance), $|D|$ is the number of patterns, x_i is the i -th pattern, c_j is the center of j -th cluster, c is the number of clusters and m is algorithm parameter – *fuzzifier*, controlling how “smooth” the transition between clusters should be.

Because Fuzzy C-means is one of the most widely known and popular fuzzy clustering algorithms, the proposed validity index was designed with this algorithm in mind and tested in concert with it. There is no apparent reason why it could not be applied to other fuzzy clustering algorithms, but its results may be different.

Validity Indices. Possibly due to importance of the problem validity indices for clustering and fuzzy clustering are already numerous.

An important group of “crisp” validity indices is based on treating data as resulting from mixture of an unknown number of distributions of certain general form (usually from exponential family) or generated by some unknown process, and attempting to select the most likely model. Such criteria include Akaike's Information Criterion [5], Consistent Akaike's Information Criterion (CAIC) [6,7],

Bayesian Information Criterion [8,7] etc. Classical Bayes laws can also be a foundation of fuzzy indices. For example Bayesian Score (BS) [9] is created by applying those laws with membership function value substituted for conditional probability of pattern belonging to a cluster.

Many fuzzy clustering validity indices follow the version of general goal of clustering stated at the beginning of this section rather directly – utilizing notion of (fuzzy) *compactness*, to describe how similar are patterns having high memberships in the same cluster, and (fuzzy) *separation*, to quantify how different the clusters are one from another. Those quantities are then combined into one criterion. Criteria belonging to this group differ mainly by their definitions of compactness and separation and include: Xie and Beni’s index (with modifications [3]), its extension by Kwon [10],

PCAES [11], indices by Fukuyama and Sugeno [10], Pakhira et al. [12], Rezaee [13] and others (e.g. [14,15]).

Some criteria, unlike most, use just memberships values, not the geometrical properties of data. Indices in this group are, for example: partition coefficient (PC) [2], its modified version by Dave (MPC) [16], partition entropy (PE) [2], validity index from [17] etc. Validity indices can be, of course, based on ideas different from those mentioned above. For example, index by Gath and Geva [10] uses hypervolume density, by Yu and Li [18] examines hessian of Fuzzy C-means objective function to check stability of clustering result, index described in [19] is based on different notion of stability – repeated bootstrapping and comparing clustering results between different sample selections.

As indicated in survey [10] and in [15], different cluster validity indices have different weaknesses and strong points. Therefore, development of new fuzzy clustering validity indices, not necessarily ideal, but with unique properties, is still needed.

The solution presented in this article, *Pattern Distances Ratio*, was not designed to rely on data being distributed according to some specific class of distributions. Unlike one of the mentioned groups, it examines data geometry. It could be described in terms of separation and compactness. The measures used are simple and quite similar to the ones often used as building blocks in many solutions ([12,3] and others mentioned e.g. in [10]), but they are applied in different manner – first calculate them for every *pattern*, combine them and *then* average it on the whole set – this is different from the solutions mentioned above, that usually compute both measures first on some group of examples (or the whole set), and combine them later. The order of those operations is important for this criterion (as is for most mentioned above) due to its form described in Sect. 2.

The criterion introduced in this article consists of a new criterion formula and a novel method for dealing with general monotonic tendency that can be observed in the basic formula in certain cases.

2 Definitions and Calculations

This method is fairly directly based on the general goal of clustering stated at the beginning of Sect. 1, though it treats it a bit differently from most indices

mentioned above. Its derivation is rather simple – the basic values computed for every *pattern* are:

1. A mean of distances to each cluster weighted by fuzzy membership to that cluster – which can be seen as fuzzy equivalent of measuring how far or dissimilar is each pattern from the clusters it belongs to.
2. A mean of distances to each cluster weighted by fuzzy membership function with operator NOT applied – which can be seen as fuzzy equivalent of measuring how far (or dissimilar) is that pattern from the clusters it does not belong to.

In accordance with the general goal, we consider patterns with lower values of **1** and higher values of **2** to be “better” assigned to clusters. Traditionally, we would like the criterion to have smaller values in case of “better” clustering, so ratio of those values is used for each pattern. We also use arithmetic mean for averaging on the whole set, which results following formula (and name Pattern Distances Ratio):

$$V_{PDR} = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{\sum_{j=1}^c \mu_{ij} \cdot d(x_i, c_j)}{\sum_{j=1}^c (1 - \mu_{ij}) \cdot d(x_i, c_j)} \cdot (c - 1), \quad (1)$$

where c is the number of clusters, $|D|$ is the size of training set, μ_{ij} is the membership function for i -th example and j -th cluster, $d(x_i, c_j)$ is the distance between i -th example and j -th cluster (for Fuzzy C-means – distance to cluster centroid) and $1-x$ is used as operator NOT.

This formula is devised under the assumption that probabilistic cluster partition is used (each $\sum_{j=1}^c \mu_{ij} = 1$, $\mu_{ij} \in [0, 1]$), the factor $c - 1$ is then just a way of normalizing mean coefficients in the denominator (because $\sum_{j=1}^c (1 - \mu_{ij}) = c - 1$). It is also required that:

$$\forall i \in \{1 \dots |D|\} \exists j \in \{1 \dots c\} d(x_i, c_j) > 0 \wedge \mu_{ij} < 1,$$

which keeps denominator from being 0 and is rather natural. Fuzzy C-means fulfills both conditions, if there are at least two distinct centers. If the second assumption could not be made, the criterion for a given pattern could, for example, be considered $+\infty$ whenever denominator is 0 and numerator is not 0, or 0 if they are both 0, though appropriateness of such substitution may depend on characteristics of used algorithm.

For comparing parameters values, clustering algorithm is run on the same sample set with each of the compared parameters values (e.g. cluster numbers) applied, V_{PDR} is computed for each result and parameters *minimizing* the criterion value are considered the best.

2.1 PDR as Cluster Number Selection Criterion

Values of both numerator and denominator of the formula described by Eq. **1** generally decrease when the number of clusters increases, but the rates of those

changes, and their boundaries are different – numerator approaches 0 when $c \rightarrow |D|$ and denominator approaches some non-zero value (for great number of uniformly distributed points hypercube line picking would be reasonable estimation). As a result, the ratios and their sums decrease to zero, as the number of cluster approaches $|D|$ and appear to have monotonic tendency, especially on low-dimensional or small datasets.

Therefore, although the formula can theoretically be applied for cluster numbers from 2 to $|D| - 1$, direct comparison of its values on wide range of cluster numbers may lead to errors. In practice, for larger datasets the number of meaningful clusters is usually much lower than the number of examples, so even rather severe (e.g. to $\sqrt{|D|}$, as used in [10]) restrictions on the maximum of tested values are not likely to cause any problems. On smaller ranges of cluster numbers – when the maximum tested number is much smaller than D – the monotonic tendency appears not to be overwhelmingly significant, so sometimes, such restrictions alone allow the solution to work properly (as seen in Sect. 3). They can also allow us to avoid the significant computational cost of additional clustering runs (especially if, like for Fuzzy C-means, the running time of algorithm increases with the number of centers). Still, in other cases they are not enough, so to improve performance of *PDR* as a cluster number selection criterion, a workaround for the monotonic tendency problem was devised.

An improved method of selecting cluster number. During development of this technique, an observation was made that the original criterion values display interesting dependence on the number of clusters even in cases where application of simple “choose the minimum” selection rule did not work well due to general monotonic tendency.

As such tendency was most visible on larger spans of clusters numbers, it seemed possible that alternative selection rule, one that would be more “local”, might work better. A very simple way to devise such rule would be to simply compare adjacent cluster numbers. For example:

1. What is the difference (in strict, mathematical sense) between this one and the next higher number? It would be best if it was minimal, preferably negative, hence breaking monotonicity.
2. What is the difference between this one and the next lower cluster number? This value should be negative (i.e. value for lower cluster number should be higher) and with highest absolute value.

To combine those two values, simple addition could be used and then, the minimum selected. However, as the rate of changes mentioned earlier is also not uniform, comparing plain differences for very different numbers of clusters has its drawbacks. This is why ranking, a common technique for increasing robustness in many contexts, was applied.

The second value could not be calculated for two clusters, so this case was treated separately. Therefore, the resulting selection procedure is:

1. Compute values $V_{PDR}(c)$ for $c = 2, \dots, max + 1$ where *max* is the maximum cluster number to be tested (lower than $|D| - 2$), $V_{PDR}(c)$ is the value of

formula from Eq. 1 for a given dataset, given algorithm, and clustering result achieved for cluster number c .

2. Compute values $diff_c = V_{PDR}(c) - V_{PDR}(c - 1)$ for $c = 3, \dots, max + 1$.
3. Sort the values $diff_c$ in increasing order, let r_c be the rank of $diff_c$ in the sorted series.
4. Compute final criterion (*differences ranking* PDR): $V_{PDR/DR}(c) = r_c - r_{c+1}$ for $c = 3, \dots, max$.
5. Tentatively select the cluster number with minimum $V_{PDR/DR}$, if there is a tie (which seems to be quite probable), select the cluster number with lower original criterion V_{PDR} , if there is still a (now improbable) tie, select lower cluster number.
6. If the original criterion value for two clusters ($V_{PDR}(2)$) is lower than for the tentatively selected one – $V_{PDR}(argmin(V_{PDR/DR}(c)))$ – or r_3 is the maximum rank (i.e. the highest increase or the smallest decrease in the criterion value occurred from 2 to 3 clusters) select 2 as number of clusters, otherwise confirm the tentative selection. Please note, that even the first condition does not require $V_{PDR}(2)$ to be the minimum.

It may be worthwhile to notice, that this procedure bears some resemblance to the technique a trained human (e.g. the author of this work) might apply when searching for a best cluster number with the help of a criterion with known general monotonic tendency.

3 Experiments

3.1 Datasets

Known benchmark datasets. These are benchmark datasets, obtained from [20], based on real world data, that were used for testing fuzzy clustering validity indices (datasets 1 - 4 in survey [10]) or at least clustering at some point. In all datasets the attributes were scaled so that all values fit in unit interval. Patterns with missing data, as well as features like class or id were removed.

1. *Breast Cancer Wisconsin (WBCD)* dataset. Nine out of 11 attributes and 683 out of 699 patterns were used for clustering. There are two classes and set is generally reported as having 2 clusters [10].
2. *Breast Cancer Wisconsin – Diagnostic (WDBC)* dataset; 569 patterns and 30 attributes were used for clustering. Two clusters are expected [10].
3. *Wine* dataset. Consists of 178 13-dimensional samples. There are three classes (each for different wine cultivar) and three clusters are expected [10].
4. *Iris* dataset. Simple and widely known dataset, having 150 instances and 4 attributes. Two of the three classes are not easily separable, so both 3 or 2 clusters is considered a good result [10].
5. *Mammal dentition* dataset. Contains 8 attributes - numbers of different tooth kinds for 66 mammals. Originally [21] it was considered a 7 cluster dataset. As all examples except for a few are centered around three distinct groups: ruminants, carnivores and rodents, 3 could also be considered a good result (especially if the metric used was not chosen specifically to this task).

Synthetic datasets. Features were scaled so that all values fit in unit interval.

1. Corner - Center 10-dimensional. There are 1000 examples and 11 clusters: one in the center of the unit hypercube, and 10 in randomly selected corners. Within each of clusters the points are picked from uniform distribution.
2. Corner - Center 3-dimensional. As the one above, but there are only 8 corners, so only 9 clusters. Number of examples stays at 1000.
3. One of datasets used in [10] entitled Example_1, has 3 clusters.
4. A three-dimensional set created by drawing from 5 Gaussian distributions with centers: (0.3,0.7,0.5), (0.3,0.6,0.3), (0.3,0.6,0.7), (0.7,0.3,0.7), (0.7,0.3,0.3), uncorrelated coordinates and standard deviation of 0.05 for every coordinate, save for third coordinate of the first distribution (deviation 0.1). 195 vectors were drawn from first distribution, 130 from others. There are 5 clusters. This is a quite standard model of data distribution.
5. Based on [4], but a cubic grid of points was added as noise: points run from 0.1 to 0.9 with step 0.1 in each dimension, which means 729 points of noise. Ideally, 5 clusters should be selected, but the noise is considerable.
6. As [5], but more examples were drawn from distributions: 390 from the first and 260 from the remaining four. Existing 5 clusters should be easier to detect than in [5].
7. Three dimensions, 8 clusters. Patterns are initially drawn from 8 normal distributions with uncorrelated coordinates, each $\sigma = 0.1$. Next, first coordinate of each pattern is multiplied by a number drawn from uniform distribution on interval [0.33...1). Coordinates of expected values of original normal distributions are either 0.2 or 0.8 (each variation is included). For each cluster, 50 patterns were created.
8. As [7], but cluster based on normal distribution with center (0.8, 0.8, 0.8) was omitted, so there are 350 patterns and 7 clusters.

3.2 Results

Following validity indices were tested on data sets mentioned above: PDR/DR – presented in this article and two of the criteria that achieved very good results in tests described in [10]: PBMF [12] and PCAES [11]. In case of PDR/DR and PCAES Fuzzy C-means algorithm was used with $m = 2$, in case of PBMF $m = 1.2$. Cluster numbers from 2 to $\sqrt{|D|}$ or to 9 (whichever was higher) were tested. Results of plain PDR with minimum value used as a selector are also included. The selected cluster numbers are in Tab. [1].

Fig. [1] displays PDR values for three datasets: WDBC, Wine, and synthetic dataset [8]. Causes of selection of 2 clusters for WDBC for both selection schemes are basically the same – as this is global minimum PDR , simple PDR selects it directly and PDR/DR overrides its first selection, as a special case for cluster number 2, described in point [6] of algorithm. This is not exactly the case with dataset Wine, where PDR/DR detects highest difference of difference ranks and plain PDR just selects global minimum. Obviously, this is also not the case with synthetic dataset [8], where selection schemes give different results. Although local

Table 1. No. of clusters: expected and selected by validation indices for benchmark datasets; abbreviations are used for real world datasets, numbers for synthetic ones; asterisk (*) means that the highest of tested cluster numbers was selected

Dataset	WBCD	WDBC	Wine	Iris	Mammal Dent.	1	2	3	4	5	6	7	8
expected	2	2	3	3 or 2	7 or 3	11	9	3	5	5	5	8	7
<i>PDR/DR</i>	2	2	3	3	3	11	9	3	5	2	5	8	7
<i>PBMF</i>	2	2	3	3	2	11	9	3	4	2	2	2	4
<i>PCAES</i>	2	2	3	2	2	11	9	3	5	3	6	6	5
<i>PDR</i> (simple)	2	2	3	3	*	11	9	*	5	*	*	*	*

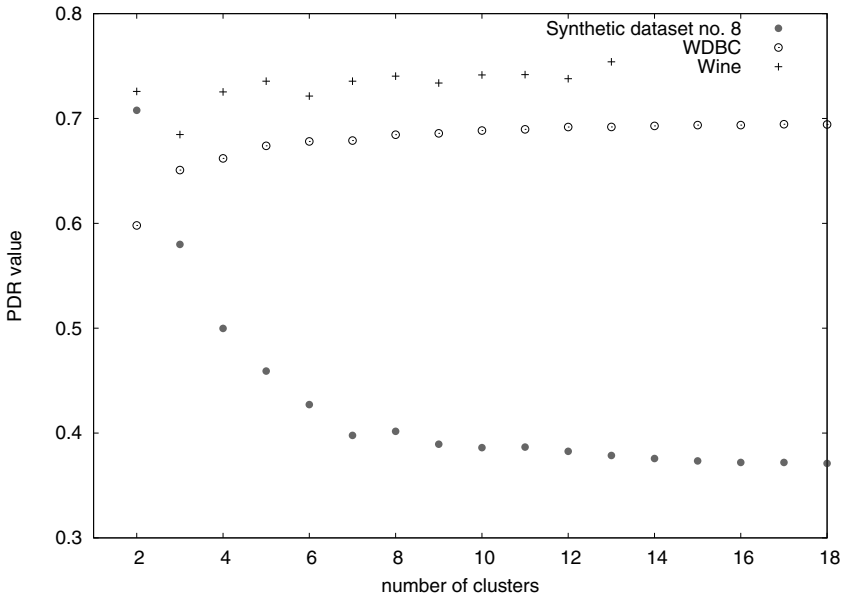


Fig. 1. Examples of values of criterion on three datasets

minima of *PDR* have usually low values of *PDR/DR*, they, in principle, might not be selected. Still, a strong local minimum will almost certainly have so low *PDR/DR* value, that it will actually be selected.

4 Conclusion

As performed tests suggest, overall performance of proposed index on real world datasets previously used in literature and synthetic datasets 1 – 6 is at least as good as the other tested criteria. Even performance of "simple" *PDR*, without using technique from Sect. 2.1, is not much worse on those datasets. Additionally the complete criterion (*PDR/DR*) performs better than other criteria on the

sets 7 and 8, representing a slightly transformed standard clustering problem. While the class of problems on which this technique outperforms the others is not yet precisely determined and might not be exceptionally wide, those good results form an advantage on top of good performance on other data, with various cluster models.

These good preliminary results may partially be the effects of quite direct translating human intuition about clustering into validity index, both at the stage of defining original validity index PDR and when developing an improved selection procedure for cluster numbers.

Because this solution computes most of the values for a given pattern, analyzing them might provide insights and possibilities hard to achieve with many other indices, such as on-line validation – as we can judge newly coming patterns separately from the whole set, but with the same measure – or detection of patterns with highest values of criterion and handling them separately.

References

1. Kruse, R., Döring, C., Lesot, M.J.: Fundamentals of fuzzy clustering. In: de Oliveira, J.V., Pedrycz, W. (eds.) *Advances in Fuzzy Clustering and its Applications*, pp. 3–30. John Wiley & Sons, Chichester (2007)
2. Bezdek, J.C.: *Pattern Recognition with Fuzzy Objective Function Algorithms (Advanced Applications in Pattern Recognition)*. Springer, Heidelberg (1981)
3. Pal, N.R., Bezdek, J.C.: On cluster validity for the fuzzy c-means model. *IEEE Transactions on Fuzzy Systems* 3(3), 370–379 (1995)
4. Brodowski, S., Podolak, I.T.: Hierarchical estimator. *Expert Systems with Applications* 38(10), 12237–12248 (2011)
5. Akaike, H.: A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19(6), 716–723 (1974)
6. Bozdogan, H.: Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika* 52, 345–370 (1987), doi:10.1007/BF02294361
7. Hu, X., Xu, L.: Investigation on several model selection criteria for determining the number of cluster. *Neural Information Processing- Letters and Reviews* 4, 2004 (2004)
8. Schwarz, G.: Estimating the dimension of a model. *The Annals of Statistics* 6(2), 461–464 (1978)
9. Cho, S.-B., Yoo, S.-H.: Fuzzy bayesian validation for cluster analysis of yeast cell-cycle data. *Pattern Recognition* 39(12), 2405–2414 (2006), *Bioinformatics*
10. Wang, W., Zhang, Y.: On fuzzy cluster validity indices. *Fuzzy Sets and Systems* 158(19), 2095–2117 (2007)
11. Wu, K.-L., Yang, M.-S.: A cluster validity index for fuzzy clustering. *Pattern Recognition Letters* 26(9), 1275–1291 (2005)
12. Pakhira, M.K., Bandyopadhyay, S., Maulik, U.: Validity index for crisp and fuzzy clusters. *Pattern Recognition* 37(3), 487–501 (2004)
13. Rezaee, B.: A cluster validity index for fuzzy clustering. *Fuzzy Sets and Systems* 161(23), 3014–3025 (2010), *Theme: Information processing*
14. Tsekouras, G.E., Sarimveis, H.: A new approach for measuring the validity of the fuzzy c-means algorithm. *Advances in Engineering Software* 35(8-9), 567–575 (2004)

15. Kim, M., Ramakrishna, R.S.: New indices for cluster validity assessment. *Pattern Recognition Letters* 26(15), 2353–2363 (2005)
16. Dave, R.N.: Validating fuzzy partitions obtained through c-shells clustering. *Pattern Recognition Letters* 17(6), 613–623 (1996)
17. Kim, Y.-I., Kim, D.-W., Lee, D., Lee, K.H.: A cluster validation index for gk cluster analysis based on relative degree of sharing. *Information Sciences* 168(1-4), 225–242 (2004)
18. Yu, J., Li, C.-X.: Novel cluster validity index for fcm algorithm. *J. Comput. Sci. Technol.* 21(1), 137–140 (2006)
19. Falasconi, M., Gutierrez, A., Pardo, M., Sberveglieri, G., Marco, S.: A stability based validity method for fuzzy clustering. *Pattern Recognition* 43(4), 1292–1305 (2010)
20. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
21. Hartigan, J.A.: Modal Blocks in Dentition of West Coast Mammals. *Systematic Biology* 25(2), 149–160 (1976)

Estimations of the Error in Bayes Classifier with Fuzzy Observations

Robert Burduk

Department of Systems and Computer Networks, Wrocław University of Technology,
Wybrzeże Wyspińskiego 27, 50-370 Wrocław, Poland
robert.burduk@pwr.wroc.pl

Abstract. The paper presents the problem of the error estimation in the Bayes classifier. The model of pattern recognition with fuzzy or exact observations of features and the zero-one loss function was assumed. For this model of pattern recognition difference of the probability of error for exact and fuzzy data was demonstrated. Received results were compared to the bound on the probability of error based on information energy for fuzzy events. The paper presents that the bound on probability of an error based on information energy is very inaccurate.

Keywords: Bayes rule, fuzzy observation, classification error.

1 Introduction

Many paper present aspect of fuzzy and imprecise information in pattern recognition [1], [2], [3], [4]. In [5] formulated the pattern recognition problem with fuzzy classes and fuzzy information and consider the following three situations:

- fuzzy classes and exact information,
- exact classes and fuzzy information,
- fuzzy classes and fuzzy information.

The classification error is the ultimate measure of the performance of a classifier. Competing classifiers can also be evaluated based on their error probabilities. Several studies have previously described the Bayes probability of error for a single-stage classifier [6], [7], for a combining classifiers [8] and for a hierarchical classifier [9], [10], [11]. Some studies pertaining to bounds on the probability of error in fuzzy concept are given in [2], [5], [12], [13].

In this paper, our aim is to obtain the error probability when the decision in the classifier is made according to the Bayes method. In our study we consider the situation with exact classes and fuzzy information on object features, i.e. when observations of the features are represented by the fuzzy sets. The received results are compared with the bound on the probability of error based on information energy.

The contents of the work are as follows. Section 2 introduces the necessary background and describes the Bayes classifier. In section 3 the basic notions of

fuzzy theory are presented. In section 4 we presented the difference between the probability of misclassification for the fuzzy and crisp data in Bayes optimal classifier.

2 Bayes Classifier

Bayesian decision theory is a fundamental statistical approach to the problem of pattern classification. This approach is based on quantifying the tradeoffs between various classification decision using probability and the costs that accompany such decision. It makes the assumption that the decision problem is posed in probabilistic terms, and that all of the probability values are known.

A pattern is represented by a set of d features, or attributes, viewed as a d -dimensional feature vector $x \in \mathfrak{R}^d$.

Let us consider a pattern recognition problem, in which the class label ω is a random variable taking values in the set of class labels $\Omega = \{\omega_1, \dots, \omega_c\}$. The *prior probabilities*, $P(\omega_i)$, $i = 1, \dots, c$ constitute the probability mass function of the variable ω , $\sum_{i=1}^c P(\omega_i) = 1$. Assume that the objects from class ω_i are distributed in $x \in \mathfrak{R}^d$ according to the *class-conditional probability density function* $p(x|\omega_i)$, $p(x|\omega_i) \geq 0$, $\forall x \in \mathfrak{R}^d$, and $\int_{\mathfrak{R}^d} p(x|\omega_i) dx = 1$, $i = 1, \dots, c$.

Given the prior probabilities and the *class-conditional probability density functions* we can calculate the *posterior probability* that the true class label of the measured x is ω_i using the Bayes formula

$$P(\omega_i|x) = \frac{P(\omega_i)p(x|\omega_i)}{p(x)} \quad (1)$$

where $p(x) = \sum_{i=1}^c P(\omega_i)p(x|\omega_i)$ is the unconditional likelihood of $x \in \mathfrak{R}^d$.

Equation (1) gives the probability mass function of the class label variable ω for the observed x . The decision for that particular x should be made with respect to the posterior probability.

The "optimal" Bayes decision rule for minimizing the risk (expected value of the loss function) can be stated as follows: Assign input pattern x to class ω_i for which the conditional risk

$$R^*(\omega_i|x) = \sum_{j=1}^c L(\omega_i, \omega_j)P(\omega_j|x) \quad (2)$$

is minimum, where $L(\omega_i, \omega_j)$ is the loss incurred in deciding ω_i when the true class is ω_j . The Bayes risk, denoted R^* , is the best performance that can be achieved. In the case of the zero-one loss function

$$L(\omega_i, \omega_j) = \begin{cases} 0, & i = j \\ 1, & i \neq j \end{cases},$$

the conditional risk becomes the conditional probability of misclassification and optimal Bayes decision rule is as follows:

$$R^*(\omega_i|x) = \max_i P(\omega_i|x). \quad (3)$$

Let Ψ^* be a classifier that always assigns the class label with the largest posterior probability. The classifier based on Bayes rule is the following:

$$\Psi^*(x) = \omega_i \quad \text{if} \quad \omega_i = \arg \max_i P(\omega_i)p(x|\omega_i). \tag{4}$$

because the unconditional likelihood $p(x) = \sum_{i=1}^c P(\omega_i)p(x|\omega_i)$ is even for every class ω_i

3 Basic Notions of Fuzzy Theory

Fuzzy number A is a fuzzy set defined on the set of real numbers \mathbb{R} characterized by means of a membership function $\mu_A(x)$, $\mu_A : \mathbb{R} \rightarrow [0, 1]$:

$$\mu_A(x) = \begin{cases} 0 & \text{for } x \leq a, \\ f_A(x) & \text{for } a \leq x \leq c, \\ 1 & \text{for } c \leq x \leq d, \\ g_A(x) & \text{for } d \leq x \leq b, \\ 0 & \text{for } x \geq b, \end{cases}$$

where f_A and g_A are continuous functions, f_A is increasing (from 0 to 1), g_A is decreasing (from 1 to 0). In special cases it may be $a = -\infty$ and (or) $b = +\infty$. In this study, the special kinds of fuzzy numbers including triangular fuzzy numbers is employed. A triangular fuzzy numbers can be defined by a triplet $A = (a_1, a_2, a_3)$. The membership function is

$$\mu_A(x) = \begin{cases} 0 & \text{for } x \leq a_1, \\ (x - a_1)/(a_2 - a_1) & \text{for } a_1 \leq x \leq a_2, \\ (a_3 - x)/(a_3 - a_2) & \text{for } a_2 \leq x \leq a_3, \\ 0 & \text{for } x \geq a_3. \end{cases}$$

The width w_A of the fuzzy number A is defined as following value [14]:

$$w_A = \int_{-\infty}^{+\infty} \mu_A(x)dx. \tag{5}$$

A fuzzy information $\mathcal{A}_k \in \mathfrak{R}^d$, $k = 1, \dots, d$ (d is the dimension of the feature vector) is a set of fuzzy events $\mathcal{A}_k = \{A_k^1, A_k^2, \dots, A_k^{n_k}\}$ characterized by membership functions

$$\mathcal{A}_k = \{\mu_{A_k^1}(x_k), \mu_{A_k^2}(x_k), \dots, \mu_{A_k^{n_k}}(x_k)\}. \tag{6}$$

The value of index n_k defines the possible number of fuzzy events for x_k (for the k -th dimension of feature vector). In addition, assume that for each observation subspace x_k the set of all available fuzzy observations (6) satisfies the orthogonality constraint [3]:

$$\sum_{l=1}^{n_k} \mu_{A_k^l}(x_k) = 1. \tag{7}$$

The probability of fuzzy event assume in Zadeh's form [15]:

$$P(A) = \int_{\mathbb{R}^d} \mu_A(x) f(x) dx. \tag{8}$$

The probability $P(A)$ of a fuzzy event A defined by (8) represents a crisp number in the interval $[0, 1]$.

4 Estimations of the Bayes Classifier Error

4.1 Estimation of the Bayes Classifier Error with Crisp Observations

The error of Ψ^* is the smallest possible error, called the Bayes error. The overall probability of error of Ψ^* is the sum of the errors of the individual x s weighted by their likelihood values $p(x)$,

$$Pe(\Psi^*) = \int_{\mathbb{R}^d} [1 - P(\omega_i^*|x)] p(x) dx. \tag{9}$$

It is convenient to split the integral into c integrals, one on each classification region. For this case class ω_i^* will be specified by the region's label. Then

$$Pe(\Psi^*) = \sum_{i=1}^c \int_{\mathbb{R}_i^*} [1 - P(\omega_i|x)] p(x) dx \tag{10}$$

where \mathbb{R}_i^* is the classification region for class ω_i , $\mathbb{R}_i^* \cap \mathbb{R}_j^* = 0$ for any $i \neq j$ and $\bigcup_{i=1}^c \mathbb{R}_i^* = \mathbb{R}^d$. Substituting (11) into (10) we have [16]:

$$Pe(\Psi^*) = 1 - \sum_{i=1}^c \int_{\mathbb{R}_i^*} P(\omega_i) p(x|\omega_i) dx. \tag{11}$$

In Fig. 1 the Bayes error is presented for the simple case of $x \in \mathbb{R}$, $\Omega = \{\omega_1, \omega_2\}$ and $P(\omega_1|x) = 1 - P(\omega_2|x)$. According to (10) the Bayes error is the area under $P(\omega_2)p(x|\omega_2)$ in \mathbb{R}_1^* plus the area under $P(\omega_1)p(x|\omega_1)$ in \mathbb{R}_2^* where $\mathbb{R}_1^* \cap \mathbb{R}_2^* = 0$. The total area corresponding to the Bayes error is marked in black.

4.2 Estimation of the Bayes Classifier Error with Fuzzy Observations

When we have non-fuzzy observation of object features in Bayes classifier then recognition algorithm for zero-one loss function is given by (3) and probability of error is given by (10). Similarly, if (7) holds and we use probability of fuzzy event given by (8) the Bayes recognition algorithm for fuzzy observations A is the following:

$$\Psi_F^*(A) = \omega_i \quad \text{if} \tag{12}$$

$$\omega_i = \arg \max_i P(\omega_i) \int_{\mathbb{R}^d} \mu_A(x) p(x|\omega_i).$$

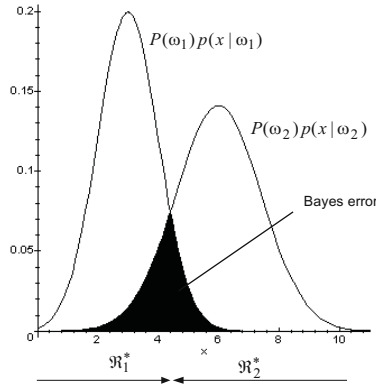


Fig. 1. The probability of error for Bayes optimal classifier when object features are non-fuzzy

The probability of error $Pe(\Psi_F^*)$ for fuzzy data is the following:

$$Pe(\Psi_F^*) = 1 - \sum_{i=1}^c \sum_{A \in i} P(\omega_i) \int_{\mathfrak{R}_i^*} \mu_A(x)p(x|\omega_i)dx, \tag{13}$$

where $A \in i$ denote the fuzzy observations belongs to the i -th classification region.

In practice we use exact or fuzzy information on object features for classification. For this two types of information the Bayes optimal error is represented by equation (11) or (13).

The difference between the probability of misclassification for the fuzzy $Pe(\Psi_F^*)$ and crisp data $Pe(\Psi^*)$ in Bayes optimal classifier is the following:

$$Pe(\Psi_F^*) - Pe(\Psi^*) = \tag{14}$$

$$= \sum_{\bar{A} \in \mathfrak{R}^d} \left(\int_{\mathfrak{R}^d} \mu_A(x) \max_i \{P(\omega_i)p(x|\omega_i)\} dx - \max_i \left\{ \int_{\mathfrak{R}^d} \mu_A(x)P(\omega_i)p(x|\omega_i)dx \right\} \right).$$

similarly as in [2].

The element of $\sum_{A \in \mathfrak{R}^d}$ equals to 0 if and only if, for the support of fuzzy observation A , one of the i discriminant functions $[P(\omega_1)p(x|\omega_1), \dots, P(\omega_i)p(x|\omega_i)]$ is uniformly larger than the others. Another interpretation is that the value of

equation (14) depends only from these observations, in whose supports intersect the discriminant functions.

4.3 Error Bounds in Terms of Information Energy for Fuzzy Observations

Some studies pertaining to bound on the probability of error in fuzzy concepts are presented in [12], [13]. They are based on information energy for fuzzy events. The information energy contained in the fuzzy event A is defined by [17]:

$$W(A) = P(A)^2 + P(\bar{A})^2, \quad (15)$$

where $P(\bar{A})$ is the complement set of A .

The information energy contained in the fuzzy information \mathcal{A} is defined by [17]:

$$W(\mathcal{A}) = \sum_{l=1}^k P(A_l)^2. \quad (16)$$

The marginal probability distribution on fuzzy information \mathcal{A} of the fuzzy event A is given by:

$$P_m(A) = \int_{\mathfrak{R}^d} \mu_A(x)p(x)dx, \quad (17)$$

where $p(x)$ is the unconditional likelihood like in (1).

The conditional information energy of Ω given by the fuzzy event A is as follows:

$$E(P(\Omega|A)) = \sum_{i=1}^c (P(\omega_i|A))^2, \quad (18)$$

where $P(\omega_i|A) = \frac{\int_{\mathfrak{R}^d} \mu_A(x)p(x|\omega_i)dx}{P_m(A)}$.

The conditional information energy of Ω given the fuzzy information \mathcal{A} is as follows:

$$E(\mathcal{A}, \Omega) = \sum_{A \in \mathcal{A}} E(P(\Omega|A))P_m(A). \quad (19)$$

For such definition of conditional information energy the upper and lower bounds on probability of error, similarly as in [13], are given by:

$$\frac{1}{2}(1 - E(\mathcal{A}, \Omega)) \leq Pe(\Psi_F^*) \leq (1 - E(\mathcal{A}, \Omega)). \quad (20)$$

4.4 Numerical Example

The aim of the experiment is to compare the estimation error of Bayesian classifiers calculated from (14) with the bounds of error classification obtained in terms of the information energy of fuzzy sets. Additionally, relationship between the

estimation error (14) and information energy of fuzzy events (16) is introduced. These results are calculated for a full probabilistic information.

Let us consider the binary classifier with *a priori* probabilities $P(\omega_1) = P(\omega_2) = 0.5$. The class-conditional probability density functions are normal distributions in \mathbb{R}^1 $p(x|\omega_1) = N(5.5, 1)$ and $p(x|\omega_2) = N(6.5, 1)$. In experiments, the following sets of fuzzy numbers were used:

case A

$$\mathcal{A} = \{A^1 = (-2, 0, 2), A^2 = (0, 2, 4), \dots, A^8 = (14, 16, 18)\},$$

case B

$$\mathcal{B} = \{B^1 = (-1, 0, 1), B^2 = (0, 1, 2), \dots, B^{16} = (14, 15, 16)\}.$$

Tab. 1 shows the difference between the probability of misclassification for fuzzy and non fuzzy data in the Bayes optimal classification calculated from (14) and information energy of fuzzy information (16) for case A. The difference $(1 - E(\mathcal{A}, \Omega)) - Pe(\Psi^*)$ for this case is equal 0.126. The change of this difference in dependence from the parameter k is the poses the precision 0.0001. The Tab. 2 shows suitable results for the case B. The information energy of fuzzy information $W(\mathcal{B})$ is equal 0.2358 and the difference $(1 - E(\mathcal{B}, \Omega)) - Pe(\Psi^*)$ is equal 0.4105 with the precision 0.0001.

The parameter k shifts the discriminant functions $P(\omega_1)p((x - k)|\omega_1)$ and $P(\omega_2)p((x - k)|\omega_2)$. Fuzzy observations are represented by adequate fuzzy numbers. The following conclusions could be drawn from the experiment:

- the difference in the misclassification for fuzzy and crisp data does not depend only on the width fuzzy number,
- the position of the class-conditional pdf's in relation to the observed fuzzy features is the essential influence for the difference $Pe(\Psi_F^*) - Pe(\Psi^*)$, i.e.

Table 1. The difference between the probability of misclassification $Pe(\Psi_F^*) - Pe(\Psi^*)$ and information energy of fuzzy information $W(\mathcal{A})$ for case A

	$p((x - k) \omega_1),$			$p((x - k) \omega_2),$		$k =$	
	0	0.5	1	1.5	2	2.5	3
$W(\mathcal{A})$	0.4093	0.4046	0.4000	0.4046	0.4093	0.4046	0.4000
$Pe(\Psi_F^*) - Pe(\Psi^*)$	0.0732	0.0390	0.0257	0.0390	0.0732	0.0390	0.0257

Table 2. The difference between the probability of misclassification $Pe(\Psi_F^*) - Pe(\Psi^*)$ for case B

	$p((x - k) \omega_1),$			$p((x - k) \omega_2),$		$k =$	
	0	0.25	0.5	0.75	1	1.25	1.5
$Pe(\Psi_F^*) - Pe(\Psi^*)$	0.0257	0.0120	0.0070	0.0120	0.0257	0.0120	0.0070

shifting the class-conditional pdf's of the parameter k increases the value of the difference $Pe(\Psi_F^*) - Pe(\Psi^*)$ up to 3.6 times for case B or up to 2.8 times for case A,

- this difference is periodical, the period is equal a half of the width fuzzy number,
- the information energy of fuzzy events is periodical too, the period is equal a half of the width fuzzy number,
- the difference $Pe(\Psi_F^*) - Pe(\Psi^*)$ is exact, the difference based on information energy $(1 - E(\mathcal{A}, \Omega)) - Pe(\Psi^*)$ is quite inaccurate estimation of the difference of error for fuzzy and crisp data.

5 Conclusion

In the present paper we have concentrated on the Bayes optimal classifier. Assuming a full probabilistic information we have presented the difference between the probability of misclassification for fuzzy and crisp data. Additionally, the received results are compared with the bound on the probability of error based on information energy.

Illustrative example shows that the position of the class-conditional probability density functions in relation to the observed fuzzy features is the essential influence for the difference $Pe(\Psi_F^*) - Pe(\Psi^*)$. Additionally, this difference is much more precisely than the difference $(1 - E(\mathcal{A}, \Omega)) - Pe(\Psi^*)$ based on information energy. The paper presents that the bound on probability of an error based on information energy is very inaccurate. This bound can be useless in a real situation of pattern recognition.

Acknowledgements. This research is supported in part by The Polish State Committee for Scientific Research under the grant which is realizing in years 2011–2013.

References

1. Burduk, R.: Decision Rules for Bayesian Hierarchical Classifier with Fuzzy Factor. In: Soft Methodology and Random Information Systems. Advances in Soft Computing, pp. 519–526 (2004)
2. Burduk, R.: Classification Error in Bayes Multistage Recognition Task with Fuzzy Observations. Pattern Analysis and Applications 13(1), 85–91 (2010)
3. Pedrycz, W.: Fuzzy Sets in Pattern Recognition: Methodology and Methods. Pattern Recognition 23, 121–146 (1990)
4. Supriya, K.D., Ranjit, B., Akhil, R.R.: An application of intuitionistic fuzzy sets in medical diagnosis. Fuzzy Sets and Systems 117(2), 209–213 (2001)
5. Okuda, T., Tanaka, H., Asai, K.: A formulation of fuzzy decision problems with fuzzy information using probability measures of fuzzy events. Information and Control 38, 135–147 (1978)
6. Antos, A., Devroye, L., Györfi, L.: Lower Bounds for Bayes Error Estimation. IEEE Trans. Pattern Analysis and Machine Intelligence 21, 643–645 (1999)

7. Avi-Itzhak, H., Diep, T.: Arbitrarily Tight Upper and Lower Bounds on the Bayesian Probability of Error. *IEEE Trans. Pattern Analysis and Machine Intelligence* 18, 89–91 (1996)
8. Woźniak, M.: Experiments on linear combiners, pp. 445–452. Springer, Heidelberg (2008)
9. Kulkarni, A.: On the Mean Accuracy of Hierarchical Classifiers. *IEEE Transactions on Computers* 27, 771–776 (1978)
10. Kurzyński, M.: On the Multistage Bayes Classifier. *Pattern Recognition* 21, 355–365 (1988)
11. Burduk, R.: The New Upper Bound on the Probability of Error in a Binary Tree Classifier with Fuzzy Information. *Neural Network World* 20(7), 951–961 (2010)
12. Pardo, L., Menendez, M.L.: Some Bounds on Probability of Error in Fuzzy Discrimination Problems. *European Journal of Operational Research* 53, 362–370 (1991)
13. Pardo, J.A., Taneja, I.J.: On the Probability of Error in Fuzzy discrimination Problems. *Kybernetes* 21(6), 43–52 (1992)
14. Chanas, S.: On the Interval Approximation of a Fuzzy Number. *Fuzzy Sets and Systems* 122, 353–356 (2001)
15. Zadeh, L.A.: Probability Measures of Fuzzy Events. *Journal of Mathematical Analysis and Applications* 23, 421–427 (1968)
16. Kuncheva, L.I.: *Combining Pattern Classifier: Methods and Algorithms*. John Wiley, New York (2004)
17. Pardo, L.: Information Energy of a Fuzzy Event and a Fuzzy Events. *IEEE Trans. on Systems, Man, and Cybernetics* SMC-15(1), 139–144 (1985)

Building Context-Aware Group Recommendations in E-Learning Systems

Danuta Zakrzewska

Institute of Information Technology Technical University of Lodz, Wolczanska 215,
90-924 Lodz, Poland
dzakrz@ics.p.lodz.pl

Abstract. Building group recommendations for students enables to suggest colleagues of similar features, with whom they can learn together by using the same teaching materials. Recommendations should depend on the context of use of an e-learning environment. In the paper, it is considered building context-aware recommendations, which aims at indicating suitable learning resources. It is assumed that learners are modeled by attributes of nominal values. It is proposed to use the method based on the Bayes formula. The performance of the technique is validated on the basis of data of students, who are described by cognitive traits such as dominant learning style dimensions. Experiments are done for real data of different groups of similar students as well as of individual learners.

Keywords: e-learning, context recommendations, Bayesian classifier.

1 Introduction

The performance of an e-learning process depends significantly on an extent to which the system is tailored to individual student needs. Students of similar preferences may form groups of peers, who can learn together from the same resources, by using educational environments of the same features. However different student characteristics may be important in accordance with course requirements or the context of the system use. Building context-aware student group recommendations can help each new student to join the suitable groups of colleagues while enrolling on different courses.

In the paper [1], the system which aimed at providing, to each new learner, recommendations of student groups of similar characteristics was considered. The system was based on agents, which were implemented to build recommendations and to indicate appropriate learning resources, or to refer the student to the tutor if a group of similar peers does not exist. It was supposed that each student features are represented by a vector of nominal values, which consist of learner cognitive styles, usability preferences or historical behaviors. It was also assumed that student models are global and the recommendations do not depend on the context of the use of an e-learning environment.

In the current paper, context based recommendations will be considered. It will be assumed, that different student features should be taken into account,

depending on the courses that a new student plans to attend. Similarly to the research described in [1], Bayes models will be applied.

The paper is organized as follows. In the next section literature review concerning recommender systems as well as application of Bayesian modeling in e-learning are presented. Then the methodology for building recommendations is described. In the following section experiments, which were carried out on real students' data as well as artificially generated data are depicted. Finally some concluding remarks are presented.

2 Related Work

In recent years many researchers examined possibilities of improving e-learning courses by using personalized recommendations of appropriate activities or learning materials. Investigations were mainly focused on the identification of learner requirements, for the purpose of suggesting matching actions or educational resources, which would support learning process [2]. Many of the researchers emphasized an importance of a context for a personalization task (see [3-5] for example). The broad review of context parameters as well as context aware e-learning systems was presented in [6].

Context-awareness was very often considered for recommendation purposes. Kim and Kwon [7] proposed the use of ontologies to create effective recommendations on the Web. Cantador et al. [8], built a news recommender system, which supports content-based and collaborative models. Andronico et al. [9] considered mobile learning process. They built a multi-agent system to suggest students educational materials taking into account learners' behavior and preferences while using different mobile devices including PDAs and cellular phones. Rosaci and Sarné [10], in turn, considered both: student's profile and an exploited device. Their recommendations were built on the basis of the time spent by the student on the particular Web site, taking into account the device used for navigating. Zaiāne [11] proposed an agent, which aims at recommending learning activities or shortcuts in course web-sites, considering learners' historical activities.

Student grouping was often used for recommendation purposes in e-learning systems. Authors grouped students according to their behaviors, taking into account pages they visited or historical navigational paths ([12, 13]), as well as learner cognitive styles or usability preferences [14]. Yang et al. [15] proposed learning resources recommendation system based on connecting similar students into small communities, where they can share resources and communicate with each other.

3 Building Group Recommendations

In a group based personalized e-learning system, each learner should be the member of the group of peers of similar traits. On the basis of group profiles the courses can be appropriately adapted. During that process, user's context

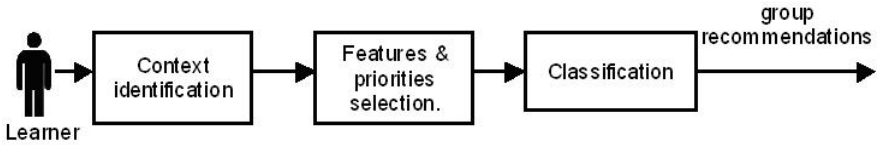


Fig. 1. Stages of recommendations' building

plays significant role. In a context-aware e-learning system, depending on a situation, different parameters should be taken into account. In order to build group recommendation to each new student the following steps should be distinguished: context identification, features and their priorities selection and finally classification (see Fig. 1). In the first step context of the system usage or of the course, on which the student intends to enroll, is identified. Then the respectful student features are chosen and their priorities are determined. Finally, after the classification process the most suitable group in the considered context is recommended.

Recommendations are built assuming that existing student groups consist of learners of similar features. They are based on three data collections: of groups' members attributes, of course resources equipped with information concerning required student features and their priorities and the third one of a new student traits. The recommender system is based on three main services: context identification, course management and classification. They are to realize the main stages of group recommendation building process. The user interacts with the highest layer, which aims at suggesting the matching group as well as respectful educational resources.

3.1 Context Identification

Context awareness in e-learning means tailoring educational system to student needs and profiles, taking into account usage circumstances. As the main factors, which decide on context parameters, there should be mentioned course requirements. They determine those student features, which should be taken into account during the process of course materials designing. We will consider those of student attributes, which values are available in both: group and user profile databases. We will also assume that priorities are assigned to all important student features appropriately to the considered course.

Let each student profile be described by a vector ST of N attributes of nominal type:

$$ST = (st_1, st_2, \dots, st_N) . \quad (1)$$

We assume that for each $i = 1, \dots, N$, st_i may take on the number of k_i nominal values. After context identification stage, let N_c attributes be selected as correlated to the considered course. Assume them to be the first ones: $st_1, st_2, \dots, st_{N_c}$. Let $w_i \geq 0, i = 1, \dots, N_c$ denote weights connected with attributes' priorities.

Those parameters will be further used in the classification stage, which aims at suggesting the best choice of group of peers to learn together during the considered course.

3.2 Classification

For the classification purpose, as the group profile we will consider its representation defined in [1]:

Definition 1. *Let GS be a group of objects described by vectors of N components of nominal type, each of which of M different values at most. As the group representation GSR we will consider the set of column vectors $gsr_i, i = 1, \dots, N$ of k_i components, representing attribute values of GS objects where $M = \max \{k_i : i = 1, \dots, N\}$. Each component of a vector $gsr_i, i = 1, \dots, N$ is calculated as likelihood $P_{ij}, i = 1, \dots, N; j = 1, \dots, k_i$ that objects from GS are characterized by the certain attribute value and is called the support for the respective attribute value in GS .*

The probabilistic form of group representations enables to use Bayes model for the classification purpose. By application of Bayesian formula, probability distribution of belonging to classes is obtained. Bayes classifier performs very well in comparison to other techniques, like decision trees, neural networks, kNN, SVM or rule-learners [19]. It ensures high accuracy of obtained results, despite of the fact that the conditional assumption, on which it is based, is rarely fulfilled in the real world. The superb performance of its classification effects was explained in the paper [20].

Bayes classifier will be used for each attribute separately, as according to our assumptions different features can have different priorities. Then:

$$P(GL_j/st_i) = \frac{P(st_i/GL_j)P(GL_j)}{\sum_{k=1}^{N_G} P(st_i/GL_k)P(GL_k)}, i = 1, \dots, N_c; j = 1, \dots, N_G; \quad (2)$$

where N_G means number of student groups, N_c is the number of considered features, $P(st_i/GL_j)$ means conditional probability and for any $j \in 1 \dots N_G$ and $i \in 1 \dots N_c$ is defined as:

$$P(st_i/GL_j) = \frac{P(st_i \cap GL_j)}{P(GL_j)}, \quad (3)$$

where GL_j is an event, which means belonging to the j -th group. st_i is a value of the i -th student attribute as presented in [1]. $P(GL_j/st_i)$ is the probability that student, whose i -th attribute is equal to st_i belongs to GL_j ; $P(GL_j)$ means probability of belonging to the group GL_j . $P(st_i/GL_j)$ is the probability that learner from GL_j is characterized by st_i .

The formula computes the probability of memberships of all of the groups, for all the considered attributes. To classify the student to the most probable group for the course, appropriate weights of the attributes should be taken into

account. We will consider application of two different techniques. In the first one, weights are taken into account during classification process and classes are indicated by the maximal value of the formula:

$$P(GL_j/ST) = \sum_{i=1}^{N_c} P(GL_j/st_i)w_i, i = 1, \dots, N_c; j = 1, \dots, N_G; \tag{4}$$

where

$$\sum_{i=1}^{N_c} w_i = 1, w_i \geq 0, i = 1, \dots, N_c. \tag{5}$$

The k-th group is recommended to the student, if

$$P(GL_k/ST) = \underbrace{\max}_{j=1, \dots, N_G} P(GL_j/ST), \tag{6}$$

and $P(GL_j/ST)$ is defined by (4), for $j = 1, \dots, N_G$.

The second technique consists on indicating the most probable group for every attribute separately, and then on choosing the group for which the majority of attributes vote. In that case, weights are taken into account in the last part of the process, while appointing the group for recommendation.

Let us assume, that suitable course resources will be prepared according to features of the majority of students in the group, then the representatives for each groups should be determined. As the group representative we will consider a vector of components equal to the nominal values for which the support is the biggest in the group (compare Def. 1). Let $R_j = (r_{j1}, r_{j2}, \dots, r_{jN})$ be the representative of the group GL_j , then for the student ST described by (1) and each group GL_j , we can define a recommendation error Err_j ; and weighted recommendation error $WErr_j, j = 1, \dots, N_G$ as follows:

$$err_{j_i} = \begin{cases} 0 & \text{if } st_i = r_{j_i}, \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

$$Err_j = \sum_{i=1}^N err_{j_i}, WErr_j = \sum_{i=1}^{N_c} w_i * err_{j_i}, \tag{8}$$

Validation of both of the techniques will be done experimentally, on the basis of the classification accuracy. We will measure it as the ratio of the number of correct suggestions to all of the recommendations. As the correct suggestion, we will mean the one, for which the value of the respective recommendation error is the smallest.

4 Learning Styles Case Study

As an example for building context aware group recommendations, student models based on learning styles will be considered. Das et al. [6] mentioned learning

styles as parameters corresponding to media used by the learner. From among different models, the one of Felder & Silverman [16] has been chosen, as it was often indicated as the most appropriate for the use in computer-based educational systems ([17]).

In the considered model, Felder & Silverman distinguished 4 attributes, which indicate preferences for 4 dimensions from among mutually excluding pairs: *active* vs. *reflective*, *sensing* vs. *intuitive*, *visual* vs. *verbal*, and *sequential* vs. *global*; or *balanced* if the student has no dominant preferences. Every dimension is responsible of separate aspects of learning. They are respectively: processing, perception, input and understanding [16]. The attribute values take the form of the odd integers from the interval $[-11,11]$, assigned for one of the dimensions from the pairs mentioned above. Each student, who filled ILS questionnaire, can be modeled by a vector SL of 4 integer attributes:

$$SL = (sl_1, sl_2, sl_3, sl_4) = (l_{ar}, l_{si}, l_{vv}, l_{sg}) , \quad (9)$$

where l_{ar} means scoring for *active* (if it has negative value) or *reflective* (if it is positive) learning style, and respectively l_{si}, l_{vv}, l_{sg} are points for all the other dimensions, with negative values in cases of *sensing*, *visual* or *sequential* learning styles, and positive values in cases of *intuitive*, *verbal* or *global* learning styles.

Score from the interval $[-3,3]$ means that the student is fairly well balanced on the two dimensions of that scale. Values $-5,-7$ or $5,7$ mean that student learns more easily in a teaching environment which favors the considered dimension; values $-9,-11$ or $9,11$ mean that learner has a very strong preference for one dimension of the scale and may have real difficulty learning in an environment which does not support that preference [18]. ST vector in that case can be presented as follows:

$$ST = SL = (ln_{ar}, ln_{si}, ln_{vv}, ln_{sg}) , \quad (10)$$

where

$$ln_{ar} = \begin{cases} a & l_{ar} = -11, -9, -7, -5, \\ b & l_{ar} = -3, -1, 1, 3, \\ r & l_{ar} = 5, 7, 9, 11; \end{cases} \quad (11)$$

$$ln_{si} = \begin{cases} s & l_{si} = -11, -9, -7, -5, \\ b & l_{si} = -3, -1, 1, 3, \\ i & l_{si} = 5, 7, 9, 11; \end{cases} \quad (12)$$

$$ln_{vv} = \begin{cases} vs & l_{vv} = -11, -9, -7, -5, \\ b & l_{vv} = -3, -1, 1, 3, \\ vr & l_{vv} = 5, 7, 9, 11; \end{cases} \quad (13)$$

$$ln_{sg} = \begin{cases} s & l_{sg} = -11, -9, -7, -5, \\ b & l_{sg} = -3, -1, 1, 3, \\ g & l_{sg} = 5, 7, 9, 11. \end{cases} \quad (14)$$

In the considered case, the group representation takes the form of the matrix and may be defined as ([1]):

Definition 2. Let GL be a cluster containing objects with ST data determined by equation (10). As the group representation we will consider the matrix $GRLR = [glr_{ij}]_{1 \leq i \leq 3, 1 \leq j \leq 4}$, where the columns represent attributes from SL model and the rows nominal values of attributes. Each element of $GRLR$ is calculated as likelihood P that students from GL are characterized by the certain attribute value from SL model and is called the support for the respective SL attribute value in GL .

$$GRLR = \begin{bmatrix} P(ln_{ar} = a), P(ln_{si} = s), P(ln_{vv} = vs), P(ln_{sg} = s) \\ P(ln_{ar} = b), P(ln_{si} = b), P(ln_{vv} = b), P(ln_{sg} = b) \\ P(ln_{ar} = r), P(ln_{si} = i), P(ln_{vv} = vr), P(ln_{sg} = g) \end{bmatrix}. \quad (15)$$

As learning styles correspond to media used for preparing and delivering learning resources [6], priorities of different dimensions may differ depending on courses, that students attend. Let us assume that for each course different weight parameters are assigned to respectful learning style dimensions. That way, each course should be equipped with a weight vector W_c of 4 components: $W_c = (w_{car}, w_{csi}, w_{c_{vv}}, w_{c_{sg}})$.

5 Experiments

The experiments were done for two different datasets of real students' attributes representing dominant learning styles of students who filled in an available online ILS self-scoring questionnaire [18] as was presented in SL model (see (10)). The process of collecting that kind of data was described with details in [21]. The first set contains data of 194 Computer Science students from different levels and years of studies, including part-time and evening courses. Those data were used for building groups of similar learners. The second set, A, contained data of students, who were to learn together with their peers from the first dataset and whose data were used for testing the recommendation efficiency. The set consists of 31 data of students studying the same master's course of Information Systems in Management, represented by 16 different SL vectors. Additionally the third set, B, of artificially generated data was used to verify the presented method. That set was formed after building student groups.

The groups of students were created as clusters of disparate structures and sizes, by application of different techniques, taking into account attributes of numeric types. Relations between numeric and nominal values of the attributes are described by (10)-(14). For the purpose of the experiments, there were considered clusters built by two well known algorithms: K-means and EM [22], taking into account schemes of 3, 4 and 5 required number of clusters. Such approach allows to consider groups of different similarity degrees and different structures. Clusters were built by using Open Source Weka software [23]. After group content analysis, 13 instances being representatives of the groups were distinguished. They constituted the set B. Classifications were done for different courses, taking into account different sets of weight values.

During the experiments, both of the proposed classification techniques were considered. Quantitative and qualitative analysis of the results were done. Quantitative analysis aimed at examining classification accuracy of the both techniques as well as the influence of course weights on the obtained results. In order to do that, different weight values for different attributes were considered. During qualitative analysis, a relation between the accuracy and the clustering schema as well as cluster sizes were investigated.

Quantitative analysis showed that, in the case of real students' data, the accuracy of weighted assignments were higher when weights were used during classification process. There were no such rules for the artificially generated data. In the last case, group representatives were chosen without taking into account weights, and including the last ones into classification process seems to lower the quality of the recommendations. For both of the considered datasets changes of weight values resulted in different group recommendations.

Table 1 presents the accuracy and exemplary weighted accuracy for both of the datasets. The first one concerns the situation, where all the features are of the same priority. Weighted accuracy is calculated for an exemplary course with weight vector equal to (0,0.1,0.6,0.3). In that case educational materials were differentiated according to input, understanding and perception, not taking into account organization of learning. The emphasis was done on adjusting visual side of the materials to student needs.

Table 1. Accuracy for the sets A and B

Set	Schema	Number of groups	Accuracy	Weight. Accuracy
A	EM	3	0.9375	1
		4	0.8125	0.875
		5	0.75	0.8125
	K-means	3	0.875	0.875
		4	0.692	0.75
		5	0.6875	0.75
B	EM	3	0.846	0.769
		4	0.692	0.615
		5	0.538	0.462
	K-means	3	0.846	0.846
		4	0.538	0.769
		5	0.846	0.769

In the case of the second technique, where the classes were chosen for each attributes separately the results were similar for the artificial dataset and worse for the set A. It seems that including weights into classification process and making suggestions for one group for all the attributes is the better choice, however to confirm that conclusion, further investigations are necessary.

Qualitative analysis showed the big influence of the group sizes on recommendations. When the differences of cluster sizes were big, Bayes classifier almost in

all of the considered cases suggested the larger group. The big impact of the lack of the representatives of certain attribute values in the group could be also noticed. In many cases the classifier did not indicate such groups as recommended even if they seemed to be the best choice from the both kinds of accuracies point of view. Finally, no correlation between accuracy and clustering schema have been noticed.

6 Concluding Remarks

In the paper, it was considered building context-aware group recommendations for students, whose preferences are characterized by nominal attributes. We assumed, that different student features should be taken into account with different weights, depending on the courses that students attend. We proposed to use group representations in the probabilistic form and consequently Bayes classifier as the main recommendation tool. The technique was examined in the case of students described by dominant learning styles. Experiments done for datasets of real students and different group structures showed that in almost all the cases the accuracies of the recommendations were very high. Tests showed that using of weights for different attributes changes classification results. It means, that recommendations should depend on course requirements.

Future research will consist in further investigations of the recommendation tool, including different computational intelligence methods, examination of other attributes, broadening the range of teaching resources and taking into account activity recommendations as well as evaluation of the significance of the proposed method by tutors.

References

1. Zakrzewska, D.: Building group recommendations in E-learning systems. In: Jędrzejowicz, P., Nguyen, N.T., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2010. LNCS, vol. 6070, pp. 391–400. Springer, Heidelberg (2010)
2. Zaïane, O.R.: Web usage mining for a better web-based learning environment. In: Proc. of Conf. on Advanced Technology for Education, Banff, AB, pp. 60–64 (2001)
3. Schmidt, A., Winterhalter, C.: User context aware delivery of e-learning material: approach and architecture. *J. Univers. Comput. Sci.* 10, 38–46 (2004)
4. Jovanović, J., Gašević, D., Knight, C., Richards, G.: Ontologies for effective use of context in e-learning settings. *Educ. Technol. Soc.* 10, 47–59 (2007)
5. Yang, S.J.H.: Context aware ubiquitous learning environments for peer-to-peer collaborative learning. *Educ. Technol. Soc.* 9, 188–201 (2006)
6. Das, M.M., Chithralekha, T., SivaSathya, S.: Static context model for context aware e-learning. *International Journal of Engineering Science and Technology* 2, 2337–2346 (2010)
7. Kim, S., Kwon, J.: Effective context-aware recommendation on the semantic web. *International Journal of Computer Science and Network Security* 7, 154–159 (2007)
8. Cantador, I., Bellogin, A., Castells, P.: Ontology-based personalised and context-aware recommendations of news items. In: Proc. of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, Australia, pp. 562–565 (2008)

9. Andronico, A., Carbonaro, A., Casadei, G., Colazzo, L., Molinari, A., Ronchetti, M.: Integrating a multi-agent recommendation system into a Mobile Learning Management System. In: Proc. of Artificial Intelligence in Mobile System 2003 (AIMS 2003), Seattle, USA, October 12 (2003)
10. Rosaci, D., Sarné, G.: Efficient personalization of e-learning activities using a multi-device decentralized recommender system. *Comput. Intell.* 26, 121–141 (2010)
11. Zaïane, O.R.: Building a recommender agent for e-learning systems. In: Proc. of the 7th Int. Conf. on Computers in Education, Auckland, New Zeland, pp. 55–59 (2002)
12. Tang, T., McCalla, G.: Smart recommendation for an evolving e-learning system. *International Journal on E-Learning* 4, 105–129 (2005)
13. Talavera, L., Gaudioso, E.: Mining student data to characterize similar behavior groups in unstructured collaboration spaces. Workshop on Artificial Intelligence in CSCL. In: 16th European Conference on Artificial Intelligence, pp. 17–23 (2004)
14. Zakrzewska, D.: Cluster analysis in personalized E-learning systems. In: Nguyen, N.T., Szczerbicki, E. (eds.) *Intelligent Systems for Knowledge Management. Studies in Computational Intelligence*, vol. 252, pp. 229–250. Springer, Heidelberg (2009)
15. Yang, F., Han, P., Shen, R., Hu, Z.: A novel resource recommendation system based on connecting to similar E-learners. In: Lau, R., Li, Q., Cheung, R., Liu, W. (eds.) *ICWL 2005. LNCS*, vol. 3583, pp. 122–130. Springer, Heidelberg (2005)
16. Felder, R.M., Silverman, L.K.: Learning and teaching styles in engineering education. *Eng. Educ.* 78, 674–681 (1988)
17. Kuljis, J., Liu, F.: A comparison of learning style theories on the suitability for elearning. In: Proc. of IASTED Conference on Web Technologies, Applications, and Services, pp. 191–197. ACTA Press (2005)
18. ILS Questionnaire, <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>
19. Kotsiantis, S.B.: Supervised machine learning: a review of classification. *Informatika* 31, 249–268 (2007)
20. Zhang, H.: The optimality of Naïve Bayes. In: Proc. of the 17th FLAIRS Conference, Florida (2004)
21. Zakrzewska, D.: Student groups modeling by integrating cluster representation and association rules mining. In: van Leeuwen, J., Muscholl, A., Peleg, D., Pokorný, J., Rumpe, B. (eds.) *SOFSEM 2010. LNCS*, vol. 5901, pp. 743–754. Springer, Heidelberg (2010)
22. Han, J., Kamber, M.: *Data Mining. Concepts and Techniques*, 2nd edn. Morgan Kaufmann Publishers, San Francisco (2006)
23. Witten, I.H., Frank, E.: *Data Mining: Practical machine learning tools and techniques*, 2nd edn. Morgan Kaufmann Publishers, San Francisco (2005)

Investigation of Random Subspace and Random Forest Methods Applied to Property Valuation Data

Tadeusz Lasota¹, Tomasz Łuczak², and Bogdan Trawiński²

¹ Wrocław University of Environmental and Life Sciences, Dept. of Spatial Management
ul. Norwida 25/27, 50-375 Wrocław, Poland

² Wrocław University of Technology, Institute of Informatics
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{tomasz.luczak,bogdan.trawinski}@pwr.wroc.pl,
tadeusz.lasota@up.wroc.pl

Abstract. The experiments aimed to compare the performance of random subspace and random forest models with bagging ensembles and single models in respect of its predictive accuracy were conducted using two popular algorithms M5 tree and multilayer perceptron. All tests were carried out in the WEKA data mining system within the framework of 10-fold cross-validation and repeated holdout splits. A comprehensive real-world cadastral dataset including over 5200 samples and recorded during 11 years served as basis for benchmarking the methods. The overall results of our investigation were as follows. The random forest turned out to be superior to other tested methods, the bagging approach outperformed the random subspace method, single models provided worse prediction accuracy than any other ensemble technique.

Keywords: random subspaces, random forest, bagging, property valuation.

1 Introduction

We have been conducting intensive study to select appropriate machine learning methods which would be useful for developing an automated system to aid in real estate appraisal devoted to information centres maintaining cadastral systems in Poland. So far, we have investigated several methods to construct regression models to assist with real estate appraisal: evolutionary fuzzy systems, neural networks, decision trees, and statistical algorithms using MATLAB, KEEL, RapidMiner, and WEKA data mining systems [11], [16], [18]. A good performance revealed evolving fuzzy models applied to cadastral data [20], [21]. We studied also ensemble models created applying various weak learners and resampling techniques [14], [17], [19].

The idea of our automated valuation system assumes a data driven modeling framework for premises valuation developed with the sales comparison method. The main advantage of data driven models is that they can be automatically generated from given datasets and, therefore, save a lot of time and financial supply. Sometimes, it is necessary to use this kind of models due to the high complexity of the process to be modeled. It was assumed that the whole appraisal area, that means the area of a

city or a district, is split into sections of comparable property attributes. The architecture of the proposed system is shown in Figure 1. The appraiser accesses the system through the internet and chooses an appropriate section and input the values of the attributes of the premises being evaluated into the system, which calculates the output using a given model. The final result, as a suggested value of the property, is sent back to the appraiser.

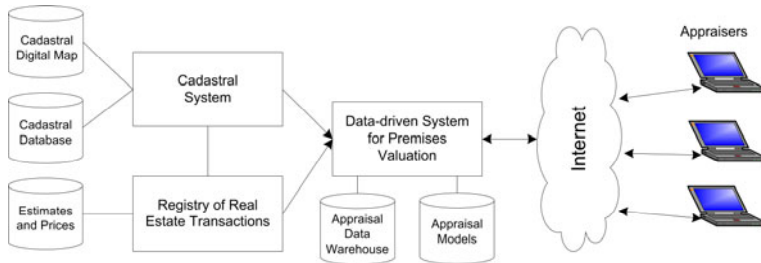


Fig. 1. Schema of automated data-driven system for property valuation

Bagging, which stands for bootstrap aggregating, devised by Breiman [2] is one of the most intuitive and simplest ensemble algorithms providing a good performance. Diversity of learners is obtained by using bootstrapped replicas of the training data. That is, different training data subsets are randomly drawn with replacement from the original training set. So obtained training data subsets, called also bags, are used then to train different classification and regression models. Finally, individual learners are combined through an algebraic expression, such as minimum, maximum, sum, mean, product, median, etc. [22]. Theoretical analyses and experimental results proved benefits of bagging, especially in terms of stability improvement and variance reduction of learners for both classification and regression problems [5], [7], [8].

Another approach to ensemble learning is called the random subspace method (RS), also known as attribute bagging [4]. It was first presented by Ho in 1995 [12]. This approach seeks learners diversity in feature space subsampling. All component models are built with the same training data, but each takes into account randomly chosen subset of features bringing diversity to ensemble. For the most part, feature count is fixed at the same level for all committee components. When it comes to classification, an ensemble makes decision either by majority voting or by weight voting. Regression is made simply by averaging components output. The method is aimed to increase generalization accuracies of decision tree-based classifiers without loss of accuracy on training data, which is one of the major problems when it comes to tree-based classifiers. Ho showed that RS can outperform bagging or in some cases even boosting [13]. While other methods are affected by the curse of dimensionality, RS can actually benefit out of it. Although it was originally designed to overcome the dilemma between overfitting and achieving maximum accuracy in tree-based classifiers, there are many studies which apply RS in pair with classifiers of different sort. Independently, Amit and Geman introduced a similar idea in the area of written character recognition [1].

Both bagging and RS were devised to increase classifier or regressor accuracy, but each of them treats the problem from different point of view. Bagging provides diversity by operating on training set instances, whereas RS tries to find diversity in feature space subsampling. Breiman [3], influenced by Amit and Geman [1], developed a method called Random Forest (RF) which merges these two approaches. RF uses bootstrap selection for supplying individual learner with training data and limits feature space by random selection. Some recent studies have been focused on hybrid approaches combining random forests with other learning algorithms [10], [15], [23].

The main goal of the study presented in this paper was to compare empirically random subspace and random forest models with bagging ensembles and single models (SM) in respect of its predictive accuracy. The algorithms were applied to real-world regression problem of predicting the prices of residential premises, based on historical data of sales/purchase transactions obtained from a cadastral system. The models were built using two weak learners including M5 model tree and multilayer perceptron neural network implemented in WEKA.

2 Methods Used and Experimental Setup

We conducted a series of experiments to compare random subspace (RS) and random forest (RF) models with bagging ensembles (BE) and single models (SM) in respect of its predictive accuracy using cadastral data on sales/purchase transaction of residential premises. All tests were accomplished using *WEKA (Waikato Environment for Knowledge Analysis)*, a non-commercial and open source data mining system [24]. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. Two following WEKA algorithms, very often used for building and exploring ensemble models, including decision tree and neural network, were employed to carry out the experiments:

M5P – Pruned Model Tree. Implements routines for generating M5 model trees. The algorithm is based on decision trees, however, instead of having values at tree's nodes, it contains a multivariate linear regression model at each node. The input space is divided into cells using training data and their outcomes, then a regression model is built in each cell as a leaf of the tree.

MLP – Multi Layer Perceptron. One of the most popular neural network. It uses backpropagation for training. In our experiment we used one hidden layer. In output layer there was only one neuron presenting prediction result.

Real-world dataset used in experiments was drawn from an unrefined dataset containing above 50,000 records referring to residential premises transactions accomplished in one Polish big city with the population of 640,000 within eleven years from 1998 to 2008. In this period most transactions were made with non-market prices when the council was selling flats to their current tenants on preferential terms. The dataset was cleansed by the experts and confined to sales transaction data of apartments built before 1997 and where the land was leased on terms of the perpetual usufruct; this form of land use is dominant in Poland. From 1997 the technology and organization of residential building construction were subject to substantial changes

in Poland and had strong impact on real estate market. Therefore, the recently built flats should be considered separately and were omitted in our study.

Next, the transactional data were combined with GIS data taken from a cadastral map and the final dataset counted 5213 records comprising nine following features pointed out by the experts as the main drivers of premises prices: usable area of premises, age of a building, number of rooms in a flat, floor on which a flat is located, number of storeys in a building, geodetic coordinates X_c and Y_c of a building, distance from the city centre and distance from the nearest shopping center. As target values total prices of premises were used, but not the prices per square meter, because the latter convey less information.

Due to the fact that the prices of premises change substantially in the course of time, the whole 11-year dataset cannot be used to create data-driven models, therefore it was split into 20 half-year subsets. Then, the prices of premises were updated according to the trends of the value changes over time. Starting from the second half-year of 1998 the prices were updated for the last day of consecutive half-years. The trends were modelled by polynomials of degree three. The sizes of half-year data subsets are given in Table 1.

Table 1. Number of instances in half-year datasets

1998-2	1999-1	1999-2	2000-1	2000-2	2001-1	2001-2	2002-1	2002-2	2003-1
202	213	264	162	167	228	235	267	263	267
2003-2	2004-1	2004-2	2005-1	2005-2	2006-1	2006-2	2007-1	2007-2	2008-1
386	278	268	244	336	300	377	289	286	181

The WEKA package enabled us to conduct the experiments within the framework of two commonly used techniques of resampling, namely 10-fold cross-validation (10cv) and holdout split into training and test sets in the proportion 70% to 30% repeated ten times (H70). The schemata of experiments with random subspace, random forest, bagging and single models within 10cv frames are shown in Figures 2, 3, 4, and 5 respectively. The schemata of tests within repeated H70 are similar.

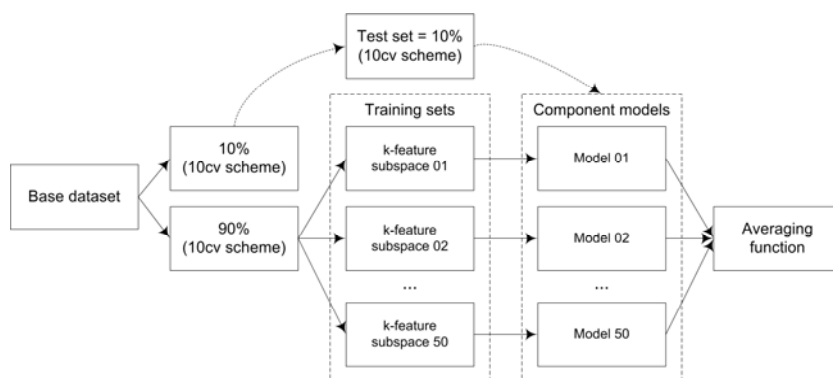


Fig. 2. Outline of experiment with random subspace method within 10cv frame. The procedure is repeated 10 times according to 10cv schema.

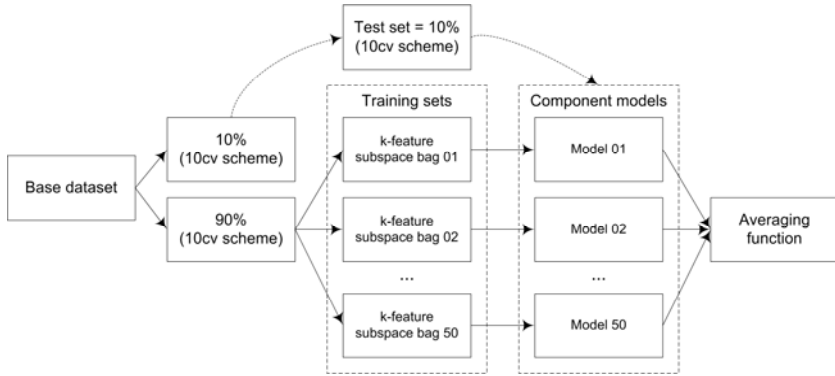


Fig. 3. Outline of experiment with random forest method within 10cv frame. The procedure is repeated 10 times according to 10cv schema.

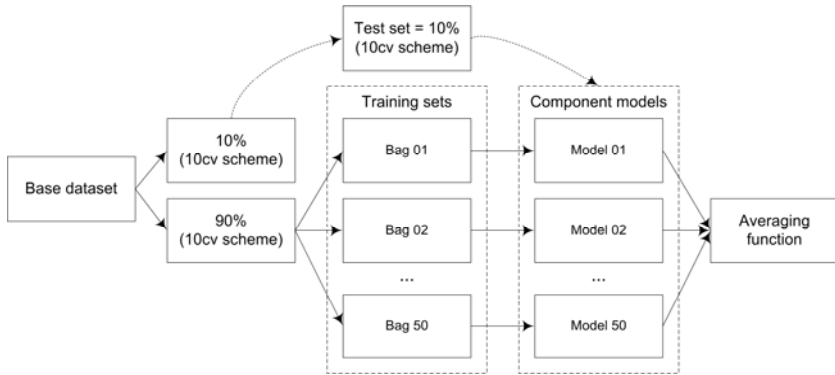


Fig. 4. Outline of experiment with bagging ensemble within 10cv frame. The procedure is repeated 10 times according to 10cv schema.

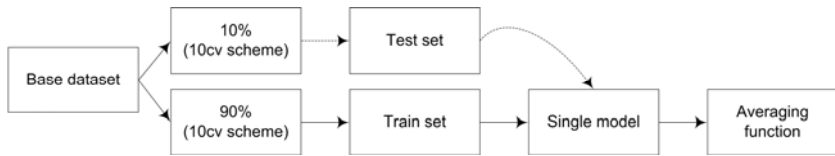


Fig. 5. Outline of experiment with single model using standard 10cv. The procedure is repeated 10 times according to 10cv schema.

All data we used in experiments were normalized using the min-max approach. As a performance function the root mean square error (RMSE) was applied. As aggregation functions averages were employed.

For verifying statistical significance of differences among all modeling methods and the expert-based technique, we conducted the pairwise non-parametric Wilcoxon test. We applied also advanced non-parametric approaches which control the error propagation of making multiple comparisons. They included the rank-based non-parametric Friedman test and its Iman-Davenport correction followed by Nemenyi's,

Holm's, Shaffer's, and Bergmann-Hommel's post-hoc procedures, which are recommended for $n \times n$ comparisons [6], [9]. We used JAVA programs available on the web page of Research Group "Soft Computing and Intelligent Information Systems" at the University of Granada (<http://sci2s.ugr.es/sicidm>).

3 Results of Experiments

The first preliminary series of experiments aimed to determine by what number of features the RS and RF models provide the best accuracy. In Figures 6 and 7 the performance of RS and RF models with different number of features for M5P and MLP algorithms within 10cv and H70 frames respectively is shown. The overall observation is as follows: the RMSE values are the highest for the smallest number of features and they drop with the increasing number of features up to 5, for 6, 7, and 8 attributes they remain at a similar level, and rise for 9 features. Moreover, there are differences between RS and RF, generally the latter outperforms the former.

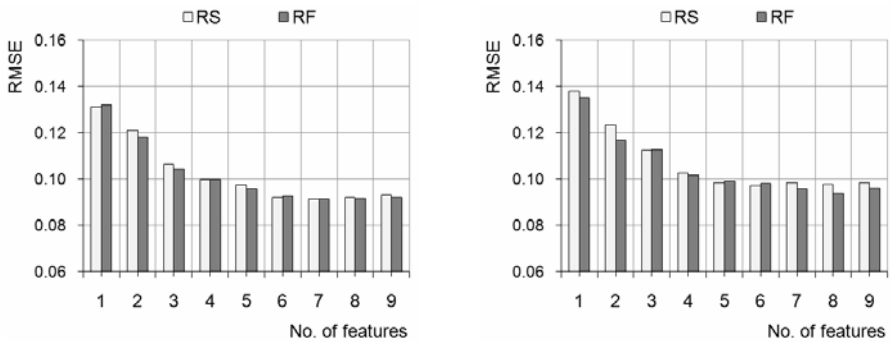


Fig. 6. Performance of *M5P* for different number of features for *10cv* (left) and *H70* (right)

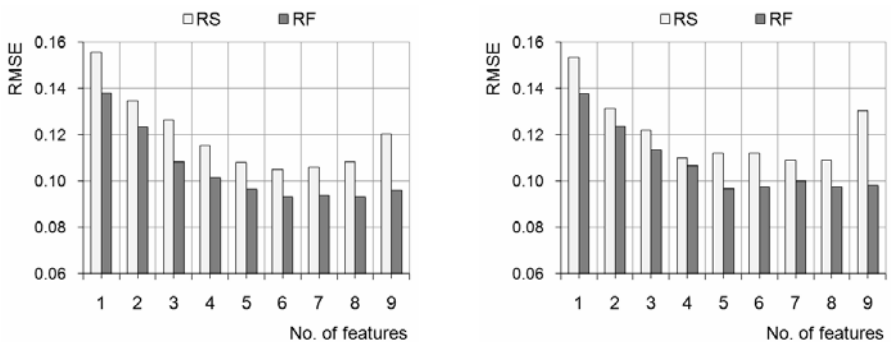


Fig. 7. Performance of *MLP* for different number of features for *10cv* (left) and *H70* (right)

The Friedman tests, performed for each combination of algorithms, ensemble method and resampling framework separately, showed that there are significant differences between models with different number of features. Average rank positions of tested models determined during Friedman test for different number of features are

presented in Table 2. Next, post-hoc procedures recommended for $n \times n$ comparisons were conducted. They revealed that models with greater number of features outperformed the models comprising up to 4 attributes. For 5 to 9 features the results were not clearly decisive. However, taking into account the rank positions to further tests we selected models with 7 features, they were marked with bold font in Table 2.

Table 2. Average rank positions of tested models determined during Friedman test for different number of features

Alg	Mth	Res	1	2	3	4	5	6	7	8	9
M5P	RS	10cv	9.00	8.00	6.90	5.95	4.75	3.10	1.90	2.15	3.25
		H70	9.00	7.60	6.40	5.00	3.90	2.65	3.30	3.30	3.85
	RF	10cv	9.00	8.00	6.90	6.00	4.95	3.50	2.15	1.95	2.55
		H70	8.95	7.65	6.95	5.20	4.05	3.85	2.40	3.25	2.70
MLP	RS	10cv	9.00	7.85	6.45	5.10	2.95	2.40	2.55	3.25	5.45
		H70	8.75	7.30	5.90	3.75	3.75	3.15	2.60	3.25	6.55
	RF	10cv	9.00	8.00	7.00	5.55	4.00	2.40	2.00	2.70	4.35
		H70	8.80	8.00	6.25	5.15	3.05	2.75	3.30	3.90	3.80

The main part of experiments was devoted to compare RS and RF models, comprising seven features randomly drawn from the whole set of nine attributes, with the BE and SM ones built over the datasets containing all 9 features. All tests were conducted for M5P and M5L algorithms within the 10cv and H70 frameworks separately. The models generated over each of 20 half-year datasets were denoted by 10cvRF7, 10cvRS7, 10cvBE9, 10cvSM9, and H70RS7, H70RF7, H70BE9, H70SM9, respectively.

For Friedman and Iman-Davenport tests showed that there were statistically significant differences among models in each test case in respect of their predictive accuracy. Average ranks of individual methods are shown in Table 3, where the lower rank value the better model. It can be seen that in each case the rank of models is the same: RF models precede the BE ones, in turn, they are before the RS ones, and finally the SM ones are preceded by all other models.

Table 3. Average rank positions of compared models determined by Friedman test

Alg	Frame	1st	2nd	3rd	4th
M5R	10cv	10cvRF7 (2.00)	10cvBE9 (2.10)	10cvRS7 (2.60)	10cvSM9 (3.30)
M5R	H70	H70RF7 (1.85)	H70BE9 (2.40)	H70RS7 (2.75)	H70SM9 (3.00)
MLP	10cv	10cvRF7 (1.20)	10cvBE9 (2.05)	10cvRS7 (2.80)	10cvSM9 (3.95)
MLP	H70	H70RF7 (1.35)	H70BE9 (1.95)	H70RS7 (2.70)	H70SM9 (4.00)

In Tables 4, 5, 6, and 7 p-values for Wilcoxon test and adjusted p-values for Nemenyi, Holm, Shaffer, and Bergmann-Hommel tests are placed for $n \times n$ comparisons for all possible 6 pairs of models. The p-values below 0.05 indicate that respective algorithms differ significantly in prediction errors (italic font). Wilcoxon test, which is devised to pairwise comparisons, rejects more zero hypotheses than post-hoc procedures worked out for $n \times n$ comparisons. Moreover, the bigger number of hypotheses is rejected for models created using MLP than M5P algorithms. For M5P within 10cv frames RF7 and BE9 performed significantly better than SM9, and within H70 frames only RF7 provided significantly lower values of RMSE than SM9.

In turn, for MLP only RF7 and BE9 produced statistically equivalent prediction errors for both frameworks, and RS7 and BE9 within H70 frames.

Table 4. Wilcoxon and adjusted p-values for $n \times n$ comparisons for M5P within 10cv frames

Alg vs Alg	pWilcox	pNeme	pHolm	pShaf	pBerg
10cvRF7 vs 10cvSM9	0.0051	0.0087	0.0087	0.0087	0.0087
10cvBE9 vs 10cvSM9	0.0152	0.0197	0.0164	0.0099	0.0099
10cvRS7 vs 10cvSM9	0.0124	0.5185	0.3456	0.2592	0.1728
10cvRS7 vs 10cvRF7	0.0569	0.8499	0.4249	0.4249	0.4249
10cvRS7 vs 10cvBE9	0.2471	1.3240	0.4413	0.4413	0.4249
10cvRF7 vs 10cvBE9	0.3507	4.8390	0.8065	0.8065	0.8065

Table 5. Wilcoxon and adjusted p-values for $n \times n$ comparisons for M5P within H70 frames

Alg vs Alg	pWilcox	pNeme	pHolm	pShaf	pBerg
H70RF7 vs H70SM9	0.0090	0.0291	0.0291	0.0291	0.0291
H70RS7 vs H70RF7	0.0080	0.1649	0.1374	0.0825	0.0825
H70BE9 vs H70SM9	0.2043	0.8499	0.5666	0.4249	0.4249
H70RF7 vs H70BE9	0.0620	1.0675	0.5666	0.5337	0.4249
H70RS7 vs H70BE9	0.6542	2.3476	0.7825	0.7825	0.4249
H70RS7 vs H70SM9	0.4553	3.2417	0.7825	0.7825	0.5403

Table 6. Wilcoxon and adjusted p-values for $n \times n$ comparisons for MLP within 10cv frames

Alg vs Alg	pWilcox	pNeme	pHolm	pShaf	pBerg
10cvRF7 vs 10cvSM9	0.0001	0.0000	0.0000	0.0000	0.0000
10cvBE9 vs 10cvSM9	0.0001	0.0000	0.0000	0.0000	0.0000
10cvRS7 vs 10cvRF7	0.0001	0.0005	0.0004	0.0003	0.0003
10cvRS7 vs 10cvSM9	0.0012	0.0291	0.0145	0.0145	0.0097
10cvRF7 vs 10cvBE9	0.0003	0.2240	0.0747	0.0747	0.0373
10cvRS7 vs 10cvBE9	0.0007	0.3972	0.0747	0.0747	0.0662

Table 7. Wilcoxon and adjusted p-values for $n \times n$ comparisons for MLP within H70 frames

Alg vs Alg	pWilcox	pNeme	pHolm	pShaf	pBerg
H70RF7 vs H70SM9	0.0001	0.0000	0.0000	0.0000	0.0000
H70BE9 vs H70SM9	0.0001	0.0000	0.0000	0.0000	0.0000
H70RS7 vs H70RF7	0.0005	0.0057	0.0038	0.0028	0.0028
H70RS7 vs H70SM9	0.0001	0.0087	0.0044	0.0044	0.0029
H70RS7 vs H70BE9	0.0064	0.3972	0.1324	0.1324	0.0662
H70RF7 vs H70BE9	0.0251	0.8499	0.1416	0.1416	0.1416

4 Conclusions and Future Work

The experiments aimed to compare the performance of random subspace and random forest models with bagging ensembles and single models in respect of its predictive accuracy were conducted using two popular algorithms M5 tree and multi layer perceptron, which belong to the so called weak learners. All tests were carried out in the WEKA data mining system within the framework of two resampling techniques, namely 10-fold cross-validation and holdout split into training and test sets in the

proportion 70% to 30% repeated ten times. A comprehensive real-world dataset including over 5200 samples and recorded during the time span of 11 years served as basis for benchmarking the methods. It was derived from cadastral data on sales/purchase transaction of residential premises accomplished in one Polish big city and combined with GIS data taken from a cadastral map. The dataset comprised nine features pointed out by the experts as main determinants of premises prices.

The overall results of our investigation were as follows. The random forest turned to be superior to other tested methods. Our study over cadastral data did not confirmed the other authors' outcome that the random subspace method outperforms the bagging approach. Single models provided worse prediction accuracy than any other ensemble technique for decision trees and neural networks tested.

It is planned to explore random subspace and random forest methods with such weak learners as genetic fuzzy systems and genetic neural networks, subsampling, and techniques of determining the optimal sizes of multi-model solutions which lead to achieve both low prediction error and an appropriate balance between accuracy and complexity. Moreover, the usefulness of the aforementioned methods to the automated system to aid in real estate appraisal will be assessed.

Acknowledgments. This work was funded partially by the Polish National Science Centre under grant no. N N516 483840.

References

1. Amit, Y., Geman, D., Wilder, K.: Joint Induction of Shape Features and Tree Classifiers. *IEEE Trans. Pattern Analysis and Machine Intelligence* 19(11), 1300–1305 (1997)
2. Breiman, L.: Bagging Predictors. *Machine Learning* 24(2), 123–140 (1996)
3. Breiman, L.: Random Forests. *Machine Learning* 45(1), 5–32 (2001)
4. Bryll, R.: Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition* 20(6), 1291–1302 (2003)
5. Bühlmann, P., Yu, B.: Analyzing bagging. *Annals of Statistics* 30, 927–961 (2002)
6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30 (2006)
7. Friedman, J.H., Hall, P.: On bagging and nonlinear estimation. *Journal of Statistical Planning and Inference* 137(3), 669–683 (2007)
8. Fumera, G., Roli, F., Serrau, A.: A theoretical analysis of bagging as a linear combination of classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(7), 1293–1299 (2008)
9. García, S., Herrera, F.: An Extension on “Statistical Comparisons of Classifiers over Multiple Data Sets” for all Pairwise Comparisons. *Journal of Machine Learning Research* 9, 2677–2694 (2008)
10. Gashler, M., Giraud-Carrier, C., Martinez, T.: Decision Tree Ensemble: Small Heterogeneous Is Better Than Large Homogeneous. In: 2008 Seventh International Conference on Machine Learning and Applications, ICMLA 2008, pp. 900–905 (2008)
11. Graczyk, M., Lasota, T., Trawiński, B.: Comparative Analysis of Premises Valuation Models Using KEEL, RapidMiner, and WEKA. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) *ICCCI 2009*. LNCS, vol. 5796, pp. 800–812. Springer, Heidelberg (2009)

12. Ho, T.K.: Random Decision Forest. In: 3rd International Conference on Document Analysis and Recognition, pp. 278–282 (1995)
13. Ho, T.K.: The Random Subspace Method for Constructing Decision Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(8), 832–844 (1998)
14. Kempa, O., Lasota, T., Telec, Z., Trawiński, B.: Investigation of bagging ensembles of genetic neural networks and fuzzy systems for real estate appraisal. In: Nguyen, N.T., Kim, C.-G., Janiak, A. (eds.) *ACIIDS 2011, Part II. LNCS (LNAI)*, vol. 6592, pp. 323–332. Springer, Heidelberg (2011)
15. Kotsiantis, S.: Combining bagging, boosting, rotation forest and random subspace methods. *Artificial Intelligence Review* 35(3), 223–240 (2010)
16. Król, D., Lasota, T., Trawiński, B., Trawiński, K.: Investigation of Evolutionary Optimization Methods of TSK Fuzzy Model for Real Estate Appraisal. *International Journal of Hybrid Intelligent Systems* 5(3), 111–128 (2008)
17. Krzystanek, M., Lasota, T., Telec, Z., Trawiński, B.: Analysis of Bagging Ensembles of Fuzzy Models for Premises Valuation. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) *Intelligent Information and Database Systems. LNCS*, vol. 5991, pp. 330–339. Springer, Heidelberg (2010)
18. Lasota, T., Mazurkiewicz, J., Trawiński, B., Trawiński, K.: Comparison of Data Driven Models for the Validation of Residential Premises using KEEL. *International Journal of Hybrid Intelligent Systems* 7(1), 3–16 (2010)
19. Lasota, T., Telec, Z., Trawiński, B., Trawiński, K.: Exploration of Bagging Ensembles Comprising Genetic Fuzzy Models to Assist with Real Estate Appraisals. In: Corchado, E., Yin, H. (eds.) *IDEAL 2009. LNCS*, vol. 5788, pp. 554–561. Springer, Heidelberg (2009)
20. Lasota, T., Telec, Z., Trawiński, B., Trawiński, K.: Investigation of the eTS Evolving Fuzzy Systems Applied to Real Estate Appraisal. *Journal of Multiple-Valued Logic and Soft Computing* 17(2-3), 229–253 (2011)
21. Lughofer, E., Trawiński, B., Trawiński, K., Lasota, T.: On-Line Valuation of Residential Premises with Evolving Fuzzy Models. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) *HAIS 2011, Part I. LNCS (LNAI)*, vol. 6678, pp. 107–115. Springer, Heidelberg (2011)
22. Polikar, R.: Ensemble Learning. *Scholarpedia* 4(1), 2776 (2009)
23. Rodríguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation Forest: A New Classifier Ensemble Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(10), 1619–1630 (2006)
24. Witten, I.H., Frank, E.: *Data Mining: Practical machine learning tools and techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Application of Data Mining Techniques to Identify Critical Voltage Control Areas in Power System

Robert A. Lis

Institute of Electric Power Engineering, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
robert.lis@pwr.wroc.pl

Abstract. Assessing and mitigating problems associated with voltage security remains a critical concern for many power system operators. It is well understood that voltage stability, is driven by the balance of reactive power in a system. Of particular interest is to identify those areas in a system that may suffer reactive power deficiencies. Establishing the reactive power reserve requirements in these Voltage Control Areas (VCAs), to ensure system integrity is of paramount importance. Since speed of analysis is critical for on-line applications the approach will address the development of a scheme whereby VCAs can be identified using data mining techniques from on-line power system snapshot. The database with user-friendly interface for storing/retrieving result of Modal Analysis can be used to construct decision trees (DTs) for each of the identified VCAs using key power system attributes. In on-line application, the relevant attributes is extracted from a system snapshot and is dropped on DTs to determine which of the pre-determined VCAs can exist in the present power system condition.

Keywords: Dynamic Security Assessment (DSA) , voltage stability, collective intelligence efficiency, decision trees.

1 Introduction

Assessing and mitigating problems associated with voltage security remains a critical concern for many power system planners and operators. Since it is well understood that voltage security is driven by the balance of reactive power in a system, it is of particular interest to find out what areas in a system may suffer reactive power deficiencies under some conditions. If those areas prone to voltage security problems, often called Voltage Control Areas (VCAs), can be identified, then the reactive power reserve requirements for them can also be established to ensure system secure operation under all conditions. A number of attempts have been made in the past to identify those areas, including a wide range of academic research and efforts toward commercial applications. A brief review of methods for determining VCA groups is presented in [2], where the author developed the Voltage Stability Security Assessment and Diagnostic (VSSAD) method. The

VSSAD method breaks up any power system into non-overlapping set of coherent bus groups (VCAs), with unique voltage stability problems. There is a Reactive Reserve Basin (RRB) associated with each VCA, which is composed of the reactive resources on generators, synchronous condensers, and other reactive power compensating devices, such that its exhaustion results in voltage instability initiated in this VCA. The VCA bus group acts like a single bus and can't obtain reactive power supply at the same level of reactive power load no matter how it is distributed among the buses in that group. Finding VCAs and their associated RRB's in VSSAD method is based on QV curve analysis performed at each test VCA. It involves the placement of a synchronous condenser with infinite limits at VCA buses and observing the reactive power generation required for different set point voltages. QV curve analysis can be time consuming if curves have to be found for every bus in the system. Finding VCA's and their associated RRB's in VSSAD method is based on QV curve analysis performed at each test VCA. It involves the placement of a synchronous condenser with infinite limits at VCA buses and observing the reactive power generation required for different set point voltages. QV curve analysis can be time consuming if curves have to be found for every bus in the system. Thus another method has been proposed by Schlueter [8,9], which reduces the number of QV curves that need to be found for determining system's RRBs. Coherent bus groups can be found by this method that have similar QV curve minima's and share a similar set of exhausted generators at these minima's. This method, however, involves a fairly high degree of trial and error and requires the computation of QV curves at higher voltage buses before the QV curves for each individual bus group can be found.

An alternative method for determining the VCA groups was proposed in [4] without the need for QV curves to be computed beforehand. The proposed sensitivity-based method ensures that buses grouped together have the same RRB generators, provided they are reactive power reserve limited. By determining which buses have similar generator branch sensitivities, it is possible to determine coherent groups of buses that will have the same RRB.

This method had been questioned in [10] based on the argument that the generator branch sensitivities are not expected to remain the same for a change in operating condition or network topology. Another method was proposed there using full Jacobian sensitivities along with bus voltage variations under contingencies.

A group of proposed methods, which are variations of the Schlueter's algorithm, rely on finding the weakest transmission lines connected to each bus. Those methods, such as Zaborszky's Concentric Relaxation method [13], are discussed to a great extent in [5]. Another method of this kind was proposed in [6] and it is based on the concept of "bus through flow". Bus static transfer stability limits are found when bus complex through flow trajectories become vertical. Those buses form topological cuts, which are connected to the rest of the system by "weak" boundaries. A Q-V sensitivity based concept of electrical distance between two buses was introduced by Lagonotte [2]. The attenuation

of voltage variation was defined as a ratio of the off-diagonal and the diagonal elements of the sensitivity matrix. Several algorithms were proposed [7][11][12] based on this concept of electrical distance for separating VCA groups. A modal analysis technique has been applied to evaluate voltage stability of large power systems [14]. Although it has proven, when combined with PV analysis, to be an effective tool for determining areas prone to voltage instability for individual selected system scenarios, it has not been used directly as an approach to automatically determine VCAs when numerous contingencies or system scenarios are involved. In summary, the existing methods have had only a limited success in commercial application because they cannot produce satisfactory results for practical systems. This, in general, is because of the following difficulties:

- *The problem is highly nonlinear:* To examine the effects of contingencies the system is repeatedly stressed in some manner by increasing system load and generation. The process of stressing the system normally introduces a myriad of non-linearities and discontinuities between the base case operating point and the ultimate instability point
- *The VCAs must be established for all expected system conditions and contingencies:* Finding VCAs is a large dimensioned problem because many system conditions and contingencies need to be considered. It may not be possible to identify a small number of unique VCAs under all such conditions. The VCAs may also change in shape and size for different conditions and contingencies.

To deal with those issues, a more practical approach is needed that can clearly establish the VCAs for a given system and all possible system conditions. The approach is based on a QV Curve method combined with Modal Analysis [3]. Typically, a QV curve is created by increasingly stressing the system and solving a power flow at each new loading point. When the power flow fails to converge, the nose of the QV curve has been reached and this point corresponds to the stability limit for that particular imposed stress. Contingencies can also be applied at points along the QV curve to generate post-contingency.

Since speed of analysis is critical for on-line applications, this work includes a scheme whereby VCAs can be identified using decision tree (DT) techniques from on-line system snapshot and a novel methodology based on the application of Data Mining (DM) techniques, where the database can be used to construct DTs (off-line) for each of the identified VCAs using key system attributes. The relevant attributes can be extracted from a system snapshot, obtained on-line, which then can be dropped on DTs to determine which VCAs can exist in the present system condition. In addition, it is proposed to investigate the possibility of predicting the required reserve using regression trees (RT) constructed (off-line) for each of the identified VCAs using key system attributes. DTs and RTs have the added benefit of identifying the most important parameters associated with a specified outcome (such as instability). This is expected to provide valuable information to power systems operators. The original contribution of the paper is to investigate and devise a methodology for identifying areas in power

systems that are prone to voltage instability (VCA's) under particular operating conditions and contingencies (which were suitable for the planning environment) and to extend this concepts to be suitable for use in the on-line (operational) environment using an intelligent system framework - techniques such as decision trees and the performance collective intelligence efficiency, to predict the VCAs and required reactive reserves from a given system condition without full computation. This is the justification for exploring the use of intelligent systems for use in on-line DSA systems.

1.1 Proposed Approach

The proposed approach is based on a QV Curve method combined with Modal Analysis. The general approach is as follows:

- A system operating space is defined based on a wide range of system load conditions, dispatch conditions, and defined transactions (source-to-sink transfers).
- A large set of contingencies is defined which spans the range of credible contingencies.
- Using QV curve methods, the system is pushed through every condition, under all contingencies until the voltage instability point is found for each condition.
- To identify the VCA for each case using modal analysis: At the point of instability for each case (nose of the PV curve) modal analysis is performed to determine the critical mode of instability as defined by a set of bus participation factors corresponding to the zero eigenvalue (bifurcation point).
- The results of the modal analysis will be placed in a database for analysis using data mining methods to identify the VCAs and track them throughout the range of system changes.
- The reactive reserve requirements for selected VCA will then be established.

The network constraints are expressed in the following linearized model around the given operating point [15]:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_{P\theta} & J_{PV} \\ J_{Q\theta} & J_{QV} \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}, \quad (1)$$

where:

ΔP – incremental change in bus real power,

ΔQ – incremental change in bus reactive power,

$\Delta\theta$ – incremental change in bus voltage angle,

ΔV – incremental change in bus voltage magnitude,

$J_{P\theta}, J_{PV}, J_{Q\theta}, J_{QV}$ - are Jacobian sub-matrices.

The elements of the Jacobian matrix give the sensitivity between power flow and bus voltage changes. While it is true that both P and Q affect system voltage stability to some degree, we are primarily interested in the dominant relationship between Q and V. Therefore, at each operating point, we may keep P constant

and evaluate voltage stability by considering the incremental relationship between Q and V . This is not to say that we neglect the relationship between P and V , but rather we establish a given P for the system and evaluate, using modal analysis, the Q - V relationship at that point. Based on the above consideration the incremental relationship between Q and V can be derived from Equation 1 by letting $\Delta P = 0$:

$$\Delta Q = J_R \cdot \Delta V . \quad (2)$$

Sensitivity matrix J_R^{-1} is a full matrix whose elements reflect the propagation of voltage variation through the system following a reactive power injection in a bus.

2 VCA Identification

In the presented approach, the power system is stressed to its stability limit for various system conditions under all credible contingencies. At the point of instability (nose of the QV curve) modal analysis is performed to determine the critical mode of voltage instability for which a set of bus participation factors (PF) corresponding to the zero eigenvalue (bifurcation point) is calculated. Based on these PFs, the proposed method identifies the sets of buses and generators that form the various VCAs in a given power system. It is assumed that for a given contingency case, buses with high PFs including generator terminal buses, form a VCA. This suggests that each contingency case might produce its own VCA. In practice, however, the large number of credible contingency cases generally will produce only a small number of VCAs because several contingencies are usually related to the same VCA. The proposed identification procedure applies heuristic rules to:

- group contingencies that are related to the same VCA;
- identify the specific buses and generators that form each VCA (see Figure II).

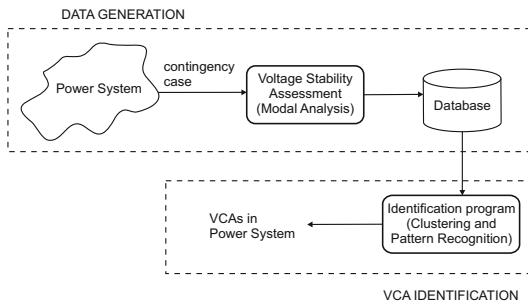


Fig. 1. VCA Identification Process

The following is a brief description of the proposed VCA identification program. The program processes the sets of buses and generators corresponding to the PFs obtained from the MA for each system condition and contingency case. Then contingency cases are grouped together if their sets of bus PFs are similar. To carry out this contingency clustering process, first a 'base/seed' set of VCA buses is selected. Then, all the other sets corresponding to different contingency cases are compared against this base set to determine if they are similar. Contingencies are clustered if their sets of bus PFs are similar. Clustering is carried out twice: clustering contingency cases based on SFAs and clustering Ck based on GENs. Each clustering process starts with the selection of a base set for the cluster. Then any other set is compared to this base to evaluate whether they are similar. Both clustering processes are specifically described and explained in detail in [16]. Finally, the program identifies the sets of buses and generators that are common to all contingencies of each cluster. Those sets of buses and generators form the VCAs of the power system.

3 Use of Decision Trees for On-Line VSA Assessment

An intelligent system framework for the application of decision trees for on-line transient stability assessment was described in [17]. A similar approach is described here for use in the assessment of voltage stability to determine VCAs and required reactive reserves that must be maintained to ensure security of each VCA. The overall architecture is shown in Figure 2.

In this Figure:

- Path D+E represents the conventional on-line VSA cycle
- Path F+G takes the conventional VSA output and computes the VCAs for all scenarios

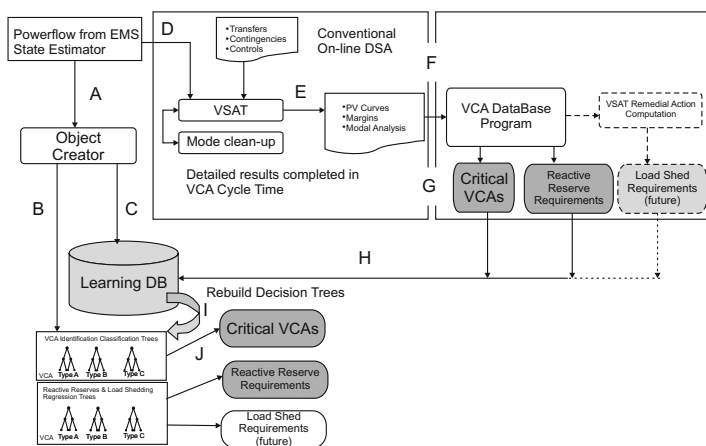


Fig. 2. Path for VCA Analysis Using Decision Trees

- Path A+C combined with Path H creates a new object for the learning database including pre-contingency power flow conditions (A+C) and the corresponding
- VCA (H). This is a learning step used to add more information to the learning database.
- Path I represents building and rebuilding of the decision trees
- Path A+B+J represents the real-time use of the intelligent system in which a power flow from the state estimator is "dropped" on the decision trees

3.1 Building the Decision Trees for VCA Determination

The objective is to develop decision trees that can be used to determine what VCA(s) may be present for a given system condition. Generally, there are two types of decision trees [18]:

- classification trees (CT) are built for classifying data samples in terms of categorical target attribute (response variable)
- regression trees (RT) have continuous target attribute (response variable).

In the data set used for decision trees, one attribute is named target attribute or goal attribute representing decisions for classification, regression or prediction, and the others are *candidate attributes* (describing pre-decision conditions).

Splitting criteria is a fundamental part of any algorithm that constructs a decision tree from a dataset is the method in which it selects attributes at each node of the tree. It may be better to place certain attributes higher up the tree in order to produce a short tree.

Some attributes split the data up more purely than others.

- their values correspond more consistently with instances that have particular values of the target attribute (the one we want to predict) than those of another attribute.
- such attributes have some underlying relationship with the target attribute.

Entropy is a sort of measure that enables us to compare attributes with each other and then be able to decide to put ones that split the data more purely higher up the tree:

- Entropy is a measure used in Information Theory and is often used in decision tree construction (def: measure of randomness and a measure of the loss of information in a transmitted signal or message)
- Informally, the entropy of a dataset can be considered to be how disordered it is.
- It has been shown that entropy is related to information, in the sense that the higher the entropy, or uncertainty, of some data, then the more information is required in order to completely describe that data.
- In building a decision tree, the aim is to decrease the entropy of the dataset until I reach leaf nodes at which point the subset that I am left with is pure, or has zero entropy and represents instances all of one class.

The entropy of a dataset S is measured with respect to one attribute, in this case the target attribute, with the following calculation:

$$Entropy(S) = - \sum_{i=1}^c p_i \log_2 p_i , \quad (3)$$

where p_i is the proportion of instances in the dataset that take the i_{th} value of the target attribute. This probability measure gives an indication of how uncertain we are about the data. A \log_2 measure is used as this represents how many bits we would need to use in order to specify what the class (value of the target attribute) is of a random instance. Using simple entropy measure can be used to recursively build a decision tree. This technique is flawed because it favors attributes that have many possible values over those that have few. Resulting tree might classify the training data which we used to build it, but as well as being complex, it won't be much good for predicting new instances. The approach to constructing decision trees usually involves using greedy heuristics (such as Entropy reduction) that can over-fit the training data and can lead to poor accuracy in future predictions. In response to the problem of over-fitting nearly all modern decision tree algorithms adopt a pruning strategy of some sort. Many algorithms use a technique known as post-pruning or backward pruning [19]. Essentially involves growing the tree from a dataset until all possible leaf nodes have been reached (i.e. purity) and then removing particular subtrees:

- At each node in a tree it is possible to see the number of instances that are misclassified on a testing set by propagating errors upwards from leaf nodes.
- This can be compared to the error-rate if the node was replaced by the most common class resulting from that node. If the difference is a reduction in error, then the subtree at the node can be considered for pruning.
- This calculation is performed for all nodes in the tree and whichever one has the highest reduced-error rate is pruned.
- The procedure is then recurred over the freshly pruned tree until there is no possible reduction in error rate at any node.

In the data set used for decision trees, one attribute is named target attribute or goal attribute representing decisions for classification, regression or prediction, and the others are candidate attributes (describing pre-decision conditions). The first step is to construct a database of the structure shown in Figure 3. Each row of the database (referred to as "objects") represents the pre-contingency conditions of a system scenario (base case condition, contingency, and transfer) for which the stability limit and VCA was found (as described in the previous section).

For each scenario, the columns of the DB (referred to as "candidate attributes") contain key power base case parameters that define the system condition, and one additional column (referred to as "target attributes") for each of the different VCAs found in the full analysis space using the VCA identification process described above.

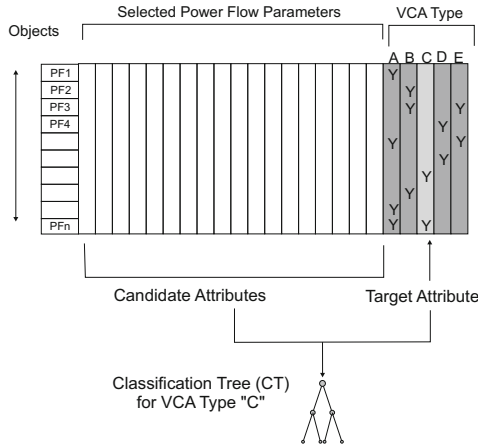


Fig. 3. Creation of Decision (Classification) Tree

The any of the VCA types occur for the given object, a "Y" (for "yes") is entered in the column, otherwise it is left blank or entered as "N" for "no". One decision tree is needed for each of the VCAs determined in the VCA grouping process. For example if VCA identification (using the VCA database) indicates five VCAs (which can be referred to as type A, B, C, D, and E, for example) five trees will be needed - one for each VCA type. When used, each DT will indicate whether a specific VCA is likely for the system condition being tested. To build a DT for a selected VCA type, the column corresponding to the VCA type is selected as the "Target Attribute" and the other VCA columns are ignored. The DT building software is then given the entire database and a DT developed for each VCA type. A similar approach is used for developing DT for reactive reserve requirements. The required reactive reserves required for each VCA are placed in the target attribute columns and new DT computed. In the case of VCA identification, the decision trees are actually *classification trees* (with a "yes" or "no" binary output) whereas the decision trees for reactive power reserve requirements are *regression trees* (with continuous outputs which indicated the actual Mvar value required for reserves). Once the trees are developed, they are made available in the on-line system as small executable programs. Graphically, a typical decision tree for a practical power system appears as shown in Figure 4.

When a snapshot is taken from the real-time system, the full on-line DSA engine is started as usual (Path D+E). In parallel, the snapshot is first formatted into a "new object" to contain the attributes used in the DB (Path A). Next, the DB is searched using a nearest neighbor (KNN) routine to ensure that the new object is within the scope of the DB; if it is not, the DT path is not used and conventional DSA is used. The new object is then "dropped" on all the decision trees (Path B) and the VCAs that are present are indicted (Path J), virtually instantaneously, as a "Yes" or "No" outcome of each of the decision trees. As the full DSA solution completes for all contingencies, the VCA database is grown

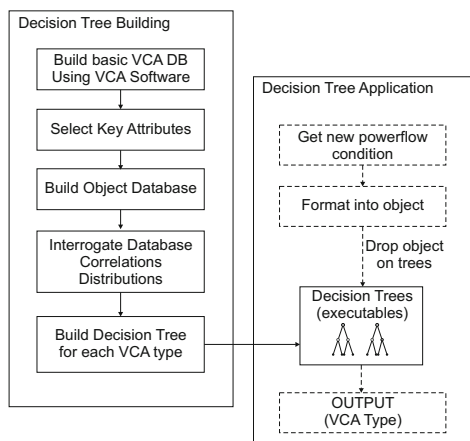


Fig. 4. Decision Tree Procedure for Determining VCA Types

by adding new calculation results and the VCA identification process can be updated at anytime as needed. Similarly, the IS database is appended with the new objects and outcomes, and new DTs can be built at anytime. This process is critical to ensure the system "learns" over time.

4 Conclusion

A highly automated method has been developed for the identification of areas prone to voltage instability (voltage control areas or VCAs) in practical power system models. For a wide range of system conditions and contingencies, the technique can identify the buses in each VCA and identify VCAs which are common for a set of contingencies and/or conditions. In addition, the method identifies the generators which are critical to maintaining stability for a given VCA. The approach was successfully implemented and tested on the Polish Power Grid Operator - PSE Operator S.A. A database with user-friendly interface for storing/retrieving result of modal analysis for each scenario/contingency was designed using MS ACCES. VCA identification was carried using a clustering method was developed. For the studied scenarios and contingencies in the PSE Operator S.A. system two VCAs were identified. For each of the identified VCA, a set of reactive resources, e.g. generators (RRG), was also identified for which the exhaustion of their reactive power reserve resulted in voltage collapse in the VCA.

References

1. Voltage Stability Assessment: Concepts: Concepts, Practices and Tools. IEEE Power Engineering Society, Power System Stability Subcommittee Special Publication (2002)

2. Schlueter, R.A.: A Voltage Stability Security Assessment Method. *IEEE Trans. Power Syst.* 13, 1423–1438 (1998)
3. Schlueter, R.A., Liu, S., Ben-Kilian, K.: Justification of the Voltage Stability Security Assessment and Diagnostic Procedure Using a Bifurcation Subsystem Method. *IEEE Trans. Power Syst.* 15, 1105–1111 (2000)
4. Aumuller, C.A., Saha, T.K.: Determination of Power System Coherent Bus Groups by Novel Sensitivity-Based Method for Voltage Stability Assessment. *IEEE Trans. Power Syst.* 18, 1157–1164 (2003)
5. Tovar, G.E., Calderon, J.G., de la Torre, V.E., Nieva, F.I.: Reactive Reserve Determination Using Coherent Clustering Applied to the Mexican Norwest Control Area. In: *Power Systems Conference and Exposition, PSCE 2004, New York City* (2004)
6. Grijalva, S., Sauer, P.W.: Static Collapse and Topological Cuts. In: *38th Annual Hawaii International Conference on System Science, Waikoloa, HI* (2005)
7. Zhong, J., Nobile, E., Bose, A., Bhattacharya, K.: Localized Reactive Power Markets Using the Concept of Voltage Control Areas. *IEEE Trans. Power Syst.* 19, 1555–1561 (2004)
8. Schlueter, R.A., Hu, I., Chang, M.W., Lo, J.C., Costi, A.: Methods for Determining Proximity to Voltage Collapse. *IEEE Trans. on Power Syst.* 6, 285–292 (1991)
9. Lie, T., Schlueter, R.A., Rusche, P.A., Rhoades, R.: Method of Identifying Weak Transmission Network Stability Boundaries. *IEEE Trans. on Power Syst.* 8, 293–301 (1993)
10. Verma, M.K., Srivastava, S.C.: Approach to Determine Voltage Control Areas Considering Impact of Contingencies. *IEE Proc.-Gener. Trans. Distrib.* 152 (2005)
11. Liu, H., Bose, A., Vencatasubramanian, V.: A Fast Voltage Security Assessment Method Using Adaptive Bounding. *IEEE Trans. Power Syst.* 15, 1137–1141 (2000)
12. Lagonotte, P., Sabonnadiere, J.C., Leost, J.Y., Paul, J.P.: Structural analysis of the electrical system: Application to secondary voltage control in France. *IEEE Trans. Power Syst.* 4, 479–485 (1989)
13. Zaborszky, J., et al.: Fast Contingency Evaluation using Concentric Relaxation. *IEEE Trans. Power Syst. PAS-99*, 28–36 (1980)
14. Gao, B., Morison, G.K., Kundur, P.: Voltage Stability Evaluation Using Modal Analysis. *IEEE Trans. Power Syst.* 7, 1529–1542 (1992)
15. Kundur, P.: *Power Systems Stability and Control*. McGraw-Hill, New York (2004)
16. Lis, R., Bajszczak, G.: Application of Voltage Control Area to Determine Reactive Power Requirements. In: *MEPS 2010, Wroclaw, Poland* (2010)
17. Rovnyak, S., Kretsinger, S.: Decision Trees For Real-Time Transient Stability Prediction. *IEEE Transactions on Power Syst.* 9 (1994)
18. Alpaydin, E.: *Introduction to Machine Learning*. MIT Press, London (2004)
19. Mehta, M., Rissanen, J., Agrawal, R.: *MDL-Based Decision Tree Pruning* (1995)

Linkage Learning Based on Local Optima

Hamid Parvin and Behrouz Minaei-Bidgoli

School of Computer Engineering,
Iran University of Science and Technology (IUST), Tehran, Iran
{parvin,b_minaei}@iust.ac.ir

Abstract. Genetic Algorithms (GAs) are categorized as search heuristics and have been broadly applied to optimization problems. These algorithms have been used for solving problems in many applications, but it has been shown that simple GA is not able to effectively solve complex real world problems. For proper solving of such problems, knowing the relationships between decision variables which is referred to as linkage learning is necessary. In this paper a linkage learning approach is proposed that utilizes the special features of the decomposable problems to solve them. The proposed approach is called Local Optimums based Linkage Learner (LOLL). The LOL algorithm is capable of identifying the groups of variables which are related to each other (known as linkage groups), no matter if these groups are overlapped or different in size. The proposed algorithm, unlike other linkage learning techniques, is not done along with optimization algorithm; but it is done in a whole separated phase from optimization search. After finding linkage group information by LOL, an optimization search can use this information to solve the problem. LOL is tested on some benchmarked decomposable functions. The results show that the algorithm is an efficient alternative to other linkage learning techniques.

Keywords: Linkage Learning, Optimization Problems, Decomposable Functions.

1 Introduction

GAs are the most popular algorithms in the category of Evolutionary Algorithms (EAs). These algorithms are widely used to solve real-world problems. However when it comes to solve difficult problems, GA has deficiencies. One of the main problems of simple GAs is their blindness and oblivion about the linkage between the problem variables. It is long time that the importance of the linkage learning is recognized in success of the optimization search. There are a lot of linkage learning techniques. Some are based on perturbation methodology, some are categorized in the class of probabilistic model building approaches and some are the techniques that adapt the linkages along with the evolutionary process by employing special operators or representations.

In this paper a new linkage learning approach, which is called LOL is proposed. The proposed algorithm as its title implies, does not fall in the above mentioned categories, but it is a linkage group identification approach which tries to identify multivariate dependencies of complex problems in acceptable amount of time and with admissible computational complexity.

2 Background

In this section, Deterministic Hill Climbers (DHC) which will be used later in our algorithm and challenging problems which are used to explain and test the proposed algorithm are described. Firstly, some terms should be defined. A partial solution denotes specific bits on a subset of string positions. For example, if we consider 100-bit binary strings, a 1 in the second position and a 0 in the seventh position is a partial solution. A building block is a partial solution which is contained in an optimum and is superior to its competitors. Each additively separable problem is composed of number of partitions each of which is called a "linkage group".

In this study, DHC [10] are used to search for local optimums. In each step, the DHC flips the bit in the string that will produce the maximum improvement in fitness value. This process can be allowed to iterate until no single bit flip produces additional movement. DHC starts with a random string.

2.1 Challenging Problems

Deficiencies of GAs were first demonstrated with simple fitness functions called deceptive functions of order k . Deception functions of order k are defined as a sum of more elementary deceptive functions of k variables. In a deceptive function the global optimum (1, ,1) is isolated, whereas the neighbors of the second best fitness solution (0, ,0) have large fitness values. Because of this special shape of the landscape, GAs are deceived by the fitness distribution and most GAs converge to (0, ,0). This class of functions has a great theoretical and practical importance. An n -bit Trap5 function has one global optimum in the string where the value of all the bits is 1, and it has $(2^{n/5}) - 1$ local optimums. The local optimums are those individuals that the values of the variables in a linkage group are either 1 or 0 (they are all 1, or they are all 0) [8].

Also another additively separable function called deceptive3 [8]. An n -bit Deceptive3 function like an n -bit Trap3 function has one global optimum in the string where the value of all the bits is 1, and it has $(2^{n/3}) - 1$ local optimums.

For yet another more challenging problem we use an additively separable function, one bit Overlapping-Trap5. An n -bit Overlapping-Trap5 function has one global optimum in the string where the value of all the bits is 1 just similar to Trap5 function, and it has $(2^{(n-1)/4}) - 1$ local optimums. The local optimums are those individuals that the values of the variables in a linkage group are either 1 or 0 (they are all 1, or they are all 0) again similar to Trap5 function [8].

2.2 Linkage Learning

There are lots of approaches in the class of linkage adaptation techniques. Linkage learning GA [2] uses a special probabilistic expression mechanism and a unique combination of the (gene number, allele) coding scheme and an exchange crossover operator to create an evolvable genotypic structure. In [4] punctuation marks are added to the chromosome representation. These bits indicate if any position on the chromosome is a crossover point or in another words, a linkage group boundary. Linkage evolving genetic operator (LEGO) [5] is another linkage adaptation strategy that in order to achieve the linkages, each gene has associated with it two Boolean

flags. These two flags determine whether the gene will link to the genes to its left and right. The two adjacent genes are assumed to be linked if the appropriate flags are both set to true. Therefore building blocks are consecutive linked genes on the chromosome.

Linkage learning is necessary when there are epistatic linkages between variables. Estimation of distribution algorithms (EDAs) are among the most powerful GAs which try to find these epistatic linkages through building probabilistic models that summarize the information of promising solutions in the current population. In another words, by using probabilistic models these algorithms seek to find linkage between variables of the problem. In each generation they find as much information as possible about the variable dependencies from the promising solutions of the population. Knowing this information, the population of the next generation is created. There are numbers of estimation of distribution algorithms which their differences are often in the model building part. Bayesian Networks and marginal product models are examples of the probabilistic models that have been used by Bayesian Optimization Algorithm (BOA) [1] and Extended Compact Genetic Algorithm (ECGA) [3]. Although EDAs scale polynomial in terms of number of fitness evaluations, the probabilistic model building phase is usually computationally expensive. Perturbation-based method, detect linkage group by injecting perturbations in the population of individuals and inspecting the fitness change caused by the perturbation. Gene expression messy genetic algorithm (gemGA) which uses transcription operator for identifying linkage group is classified in this category.

Table 1. Some of the local optimums for Trap3 size 12

1	1	1	0	0	0	0	0	0	1	1	1
0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	1	1	1

Dependency Structure Matrix Genetic Algorithm (DSMGA) [8] is another approach which models the relationship among variables using Dependency Structure Matrix (DSM). In DSMGA a clustering method is used for identifying the linkage groups. In spite of these efforts, none of the algorithms have been claimed to be stronger than Hierarchical Bayesian Optimization Algorithms (HBOA) [6] and [7] which itself in spite of its polynomial scalability in terms of the number of fitness evaluations, is computationally expensive.

3 Local Optimum Based Linkage Learner: LOLL

The main idea in the proposed approach for identifying the multivariate dependencies is using local optimums. But how the local optimums can lead us to identification of the linkage groups?

Local optimums of "Additively Separable problems" have some unique features. As it is obvious, the global optimum of an additively separable problem is the one that all of its building blocks are identified. In another words, in the global optimum, all of the linkage groups of the problem have the highest possible contribution to the overall fitness. But in local optimum solutions, not all of the building blocks are found and those partitions or sub-problems of the problem which their optimum values are not found are occupied by the best competitors of the superior partial solution.

In additively separable problems there are lots of these local optimum solutions. Actually number of such solutions is directly dependant on length of the problem and number of partitions (or sub-problems or linkage groups) of the problem. It can be said that each local solution contains at least one building block (except the one with all 0s) and therefore comparison of the optimum solutions can lead us to identification of the linkage groups.

The following example can reveal this concept more clearly. Consider a 12 bit Trap3 function. This function has one global optimum 1111111111 and $(2^{12/3}) - 1 = (15)$ local optimums. The strings are local optimum if the bits corresponding to each trap partition are equal, but the value of all the bits in at least one trap partition is 0. Some of local optimums are shown in table 1: A simple comparison between first local solution and fifth local solution helps us find the second linkage group and comparison between third local solution and fourth local solution helps us find the first linkage group.

Now, the algorithm can be explained and the example is continued later. In an overall view, there are two phases of search and analysis. In search phase some local optimums are found and in analysis phase the comparisons between these local solutions are done. If number of local solutions is not enough to discover all the linkage groups of the problem, the local solutions for the remained bits of the problem that are not assigned to a linkage group yet are to be found by the comparison of the newly found local optimums. This process repeats until remained undiscovered linkage groups of the problem are identified. The process will end if all the variables of the problem are assigned to a linkage group. In the search phase, K DHCs are initialized randomly and set to search the landscape (with length (Xs) number of variables). When each DHC finds a peak in the landscape and no movements are possible, that solution which is a local optimum will be saved in a set named *HighModals*.

After the search phase, analysis phase starts. In the analysis phase, linkage groups should be identified by comparing different local optimum solutions.

A comparison method is needed for the analysis phase. The comparison method should be able to segregate the BBs of the local solutions and yet be simple and uncomplicated. XOR operation is a good candidate for this purpose. This is due to the fact that the local and global solutions of a decomposable function are the two strings with the most differences in their appearance and binary strings are used to code the individuals.

Therefore in the analysis phase, each two local optimum solutions in the *HighModals* set are *XORed* with each other and the results are stored in *XORed* set.

Therefore *XORed* is an array of arrays. The strings with least number of ones are found. Number of 1s (r) in these strings is considered the length of linkage group. And these strings (string with length r) are put in the set *DiscoveredBBs* which is an array of arrays and contains the ultimate results (all the identified linkage groups). All of the other members of *XORed* set with more than r 1s are put in the set *XsArray* which is again array of arrays. After identifying some of the linkage groups, the algorithm is recursively called for finding linkage groups in other parts of the string which are not identified yet. The undiscovered parts are the *XORed* strings which their length is more than r or those variables of the problem which are not in the *XORed* set. Therefore those bits in the *Xs* which are not in the *XORed* set are added as a separate member to the *XsArray* (step A.4 in the algorithm).

As it is mentioned before we need a mechanism to balance the time spent in the search phase. For this reason a parameter, sp is contrived which determines when to leave the search phase. Leaving the search phase takes place with the probability sp . If sp is small, the set *HighModals* will become bigger because the search phase takes longer and as a consequence more local solutions are found. Analysis of huge number of solutions is difficult and unnecessary. On the other hand by comparison of too few local solutions there is a little chance of identifying all the linkage groups of the problem. So sp parameter should be determined wisely considering the length of the problem. If the length of the problem is more, the number of local solutions needed to identify the linkage groups is more. If each variable of the problem is assigned to at least one linkage group, the LOLL algorithm terminates. The pseudo code of LOLL algorithm is shown in Fig. 1.

Xs is an array with length n containing the indexes of the problem variables. *DiscoveredBBs* is an array of arrays, containing the discovered linkage groups. Each linkage group is shown with an array containing the indexes of the variables in the linkage group. *HighModals* is an array containing the local optimums of the problem. *XORed* is an array of arrays containing the result of XOR operation on local solutions. Each XOR result is shown with an array containing the indexes of bits which their value is 1 after doing XOR operation. *DeterminedBits* is an array which contains the indexes of the variables which their corresponding linkage group is identified. *XsArray* is an array of arrays containing those parts which should be searched again for identification of the remaining linkage groups.

As it is obvious, the only parameter which should be set wisely is sp . In the future work, we address solutions to adjust this parameter automatically. Complexity of the algorithm will be discussed later. Now, we go back to our simple example: *Xs* is here the array $Xs = \{1, 2, \dots, 12\}$ *HighModals* set is in Table 1.

XORed set of our simple example: [1,2,3,4,5,6] , [1,2,3,7,8,9] , [7,8,9,10,11,12] , [4,5,6] , [1,2,3,7,8,9,10,11,12] , [1,2,3]

DiscoveredBBs set so far: [4,5,6] , [1,2,3]

DeterminedBits set: [1,2,3,4,5,6]

XsArray: [1,2,3,7,8,9] , [7,8,9,10,11,12] , [1,2,3,7,8,9,10,11,12]

LOLL algorithm is again called for three sub-problems in *XsArray*.

```

Xs=(1...n);
r=0;
DiscoveredBBs=LOL(Xs)
Search Phase:
S.1. Run DHCs;
S.2. Save local solutions to HighModals set.
S.3. If p--exit--search < (sp) goto Analysis phase.
    Else goto S.1.
Analysis Phase:
A.1. Perform XOR between each two members of HighModals set.
    Put the XORed into XORed set.
A.2. Find the strings with least number of 1s in XORed set.
    Set its number of 1s = r as length of BB, and put those XORed members
    to set DiscoveredBBs.
A.3. Delete those XORed members which their length is equal to length Xs.
A.4. Put all the indexes of bits of each member of XORed set which its length
    is equal to r into a set DeterminedBits.
A.5. For each member of XORed set i,
    If((length(i)>r)^all of the bits of i which are not in DeterminedBits)
        Put that in the set XsArray;
A.6. Put the (Xs-(indexes of all the bits which value 1 in the XORed set)) into
    XsArray;
A.7. If all of the variables of the problem ∈ DeterminedBits
    \*all of the variables are assigned to linkage group (at least) *\
        Terminates the algorithm.
Else
    for i=1 to length XsArray Do
        DiscoveredBBs=LOL(XsArray[i]);
    
```

Fig. 1. Local Optimum based Linkage Learning Algorithm

Table 2. The local optimums of Table 1. Each gene is considered as a data point and each high modal is considered as a feature

HighModals /Bits	gene ₁	gene ₂	gene ₃	gene ₄	gene ₅	gene ₆	gene ₇	gene ₈	gene ₉	gene ₁₀	gene ₁₁	gene ₁₂
highmodal ₁	1	1	1	0	0	0	0	0	0	1	1	1
highmodal ₂	0	0	0	1	1	1	1	1	1	0	0	0
highmodal ₃	0	0	0	1	1	1	0	0	0	1	1	1
highmodal ₄	1	1	1	1	1	1	0	0	0	0	0	0
highmodal ₅	1	1	1	1	1	1	0	0	0	1	1	1

3.1 Finding Local Optimums

If LOL faces with the overlapping problem, it may crash. It will also be weak, if it does not crash. In more detail, it will crash, if there are many local optimums in the environment. How can LOL face with these problems?

Again the algorithm has first phase of previous LOLL. After completing first phase we obtain a lot of local optimums, e.g. look at Table 2. In Table 2 Each gene is considered as a data point and each high modal is considered as a feature. After the search phase, clustering phase starts. In the clustering phase, linkage groups should be identified by a single clustering method.

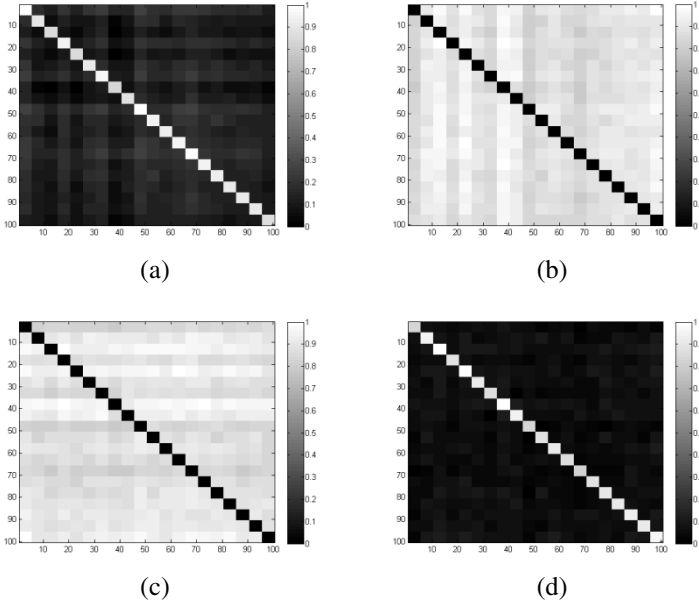


Fig. 2. (a) $Similarity_{i,j}^{0,0}$ for trap5 size 100. (b) $Similarity_{i,j}^{0,1}$ for trap5 size 100. (c) $Similarity_{i,j}^{1,0}$ for trap5 size 100. (d) $Similarity_{i,j}^{1,1}$ for trap5 size 100.

3.2 Extracting Real Building Blocks Out of Local Optimums

As it is said, if we have two or more optimums in each building block, the method has some drawbacks in finding the final *DiscoveredBBs*. For handling this drawback we use each of the found *HighModals* as a dataset. We define co-association matrix $Sim_{i,j}^{k,q}[p]$, where $k, q \in \{0, 1\}$, to be one if and only if i -th gene of p -th modal in *HighModals* be k and j -th gene of p -th modal in *HighModals* be q . After that we define matrix $S_{i,j}^{k,q} = \sum_p Sim_{i,j}^{k,q}[p]$. Now we define similarity matrix $Similarity_{i,j}^{k,q} = S_{i,j}^{k,q} / \max(S_{i,j}^{k,q})$ and dissimilarity matrix $Dissimilarity_{i,j}^{k,q} = 1 - Similarity_{i,j}^{k,q}$. For further explanation consider Fig. 2. Fig. 2 depicts the similarity matrix of a trap5 size 100. Note that each of these four matrices can represent the necessary linkages completely. Fig. 3 also depicts the dissimilarity matrix of a deceptive5 size 100. Fig. 4 shows the effectiveness of the dissimilarity (similarity) matrix in representing the linkages of overlapping traps and deceptive functions completely and meaningfully.

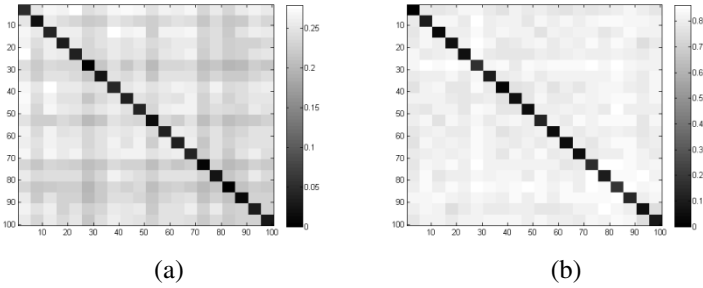


Fig. 3. (a) $Dissimilarity_{i,j}^{0,0}$ for deceptive5 size 100. (b) $Dissimilarity_{i,j}^{1,1}$ for deceptive5 size 100.

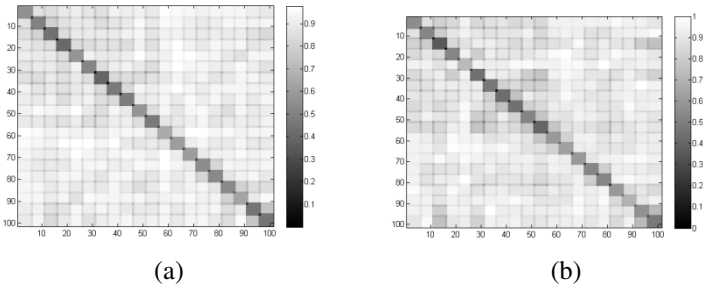


Fig. 4. (a) $Dissimilarity_{i,j}^{1,1}$ for one bit overlapping deceptive6 size 101. (b) $Dissimilarity_{i,j}^{1,1}$ for one bit overlapping trap6 size 101.

Then considering the similarity or dissimilarity matrix as a new data space of genes, a clustering algorithm can be employed to partition them. This can be done by cutting a minimal number of clusters (hyperedges) using approach the HyperGraph-Partitioning Algorithm (HGPA) [9].

4 Empirical Results

For all tested problems, 30 independent runs are performed and our approach is required to find all the linkage groups accurately in all the 30 runs. The performance of LOLL is measured by the average number of fitness evaluations until it terminates. The results of LOLL are summarized in the Fig 5. All this results are obtained without applying the clustering algorithm. As it is obvious, the finding of building blocks is sub-quadratic either in non-overlapping challenging problems, or in overlapping ones. It is worthy to note that the time order of the algorithm in the challenging problems increases as the size of building blocks increases no matter it is overlapping functions. This is very important result, because as the size of building blocks in the BOA and in the HBOA, the times orders of these algorithms increase exponentially [8].

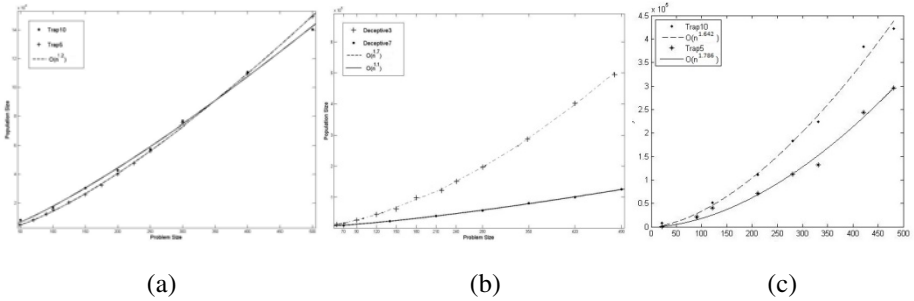


Fig. 5. (a) Number of fitness evaluations vs. problem size for trap5 and trap10. (b) Number of fitness evaluations vs. problem size for deceptive3 and deceptive7. (c) Number of fitness evaluations vs. problem size one-bit overlapping trap5 and trap10.

5 Conclusions

With the purpose of learning the linkages in the complex problem a novel approach is proposed. There are other approaches that are claimed to be able to solve those challenging problems in tractable polynomial time. But the proposed approach does not classified into the existence categories. This work has looked at the problem from whole different points of view. Our method is based on some properties of additively decomposable problems in order to identify the linkage groups. The amazing property of additively decomposable problems that our method is based on is the special form of their local optimums which a bunch of them would give us lots of information about the linkage groups. The proposed algorithm is called LOLL. The algorithm is capable of solving the challenging problems effectively.

LOLL is capable of identifying the linkage groups in a simple and straightforward manner. As it is shown in terms of numbers of fitness evaluation the complexity of LOLL has been $O(n^{1.2})$ in the two test cases over a trap problem and $O(n^{1.7})$ and $O(n^{1.1})$ in deceptive3 and deceptive7 problems. Moreover we believe that the proposed algorithm (without any major changes) is capable of finding the overlapping building blocks. The result testing the proposed approach on overlapping problems and more detailed analysis of the algorithm will be represented in our future work. Analyzing the proposed algorithm in the context of optimization problem and along with an optimization search is one of the tasks that can be done as future works. Comparing the results with the other approaches is also left as future work.

References

1. Audebert, P., Hapiot, P.: Effect of powder deposition. J. Electroanal. Chem. 361, 177 (1993)
2. Newman, J.: Electrochemical Systems, 2nd edn. Prentice-Hall, Englewood Cliffs (1991)
3. Hillman, A.R.: Electrochemical Science and Technology of Polymers, vol. 1, ch. 5. Elsevier, Amsterdam (1987)
4. Miller B.: Geelong, Vic., February 19-24; J. Electroanal. Chem., 168 (1984)

5. Jones: personal communication (1992)
6. Pelikan, M., Goldberg, D.E.: Escaping hierarchical traps with competent genetic algorithms. In: Genetic and Evolutionary Computation Conference, GECCO, pp. 511–518 (2001)
7. Pelikan, M., Goldberg, D.E.: A hierarchy machine: Learning to optimize from nature and humans. *Complexity* 8(5) (2003)
8. Pelikan, M.: Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithms. Springer, Heidelberg (2005)
9. Strehl, A., Ghosh, J.: Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research* 3, 583–617 (2002)
10. Stuart, R., Peter, N.: *Artificial Intelligence: A Modern Approach*, 2nd edn., pp. 111–114. Prentice-Hall, Englewood Cliffs (2003)

Data Extrapolation and Decision Making via Method of Hurwitz-Radon Matrices

Dariusz Jakóbczak

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

Abstract. Computational Collective Intelligence needs suitable methods of data extrapolation and decision making. Proposed method of Hurwitz-Radon Matrices (MHR) can be used in extrapolation and interpolation of curves in the plane. For example quotations from the Stock Exchange, the market prices or rate of a currency form a curve. This paper contains the way of data anticipation and extrapolation via MHR method and decision making: to buy or not, to sell or not. Proposed method is based on a family of Hurwitz-Radon (HR) matrices. The matrices are skew-symmetric and possess columns composed of orthogonal vectors. The operator of Hurwitz-Radon (OHR), built from these matrices, is described. Two-dimensional data are represented by the set of curve points. It is shown how to create the orthogonal and discrete OHR and how to use it in a process of data foreseeing and extrapolation. MHR method is interpolating and extrapolating the curve point by point without using any formula or function.

Keywords: computational intelligence, decision making, knowledge representation, curve interpolation, data extrapolation, value anticipation, Hurwitz-Radon matrices.

1 Introduction

A significant problem in computational intelligence and knowledge representation [1] is that of appropriate data representation and extrapolation. Two-dimensional data can be treated as points on the curve. Classical polynomial interpolations and extrapolations (Lagrange, Newton, Hermite) are useless for data anticipation, because the stock quotations or the market prices represent discrete data and they do not preserve a shape of the polynomial. Also Richardson extrapolation has some weak sides concerning discrete data. This paper is dealing with the method of data foreseeing by using a family of Hurwitz-Radon matrices. The quotations, prices or rate of a currency, represented by curve points, consist of information which allows us to extrapolate the next data and then to make a decision [2].

If the probabilities of possible actions are known, then some criteria are to apply: Laplace, Bayes, Wald, Hurwicz, Savage, Hodge-Lehmann [3] and others [4]. But in this paper author considers only two possibilities: to do something or not. For example to buy a share or not, to sell a currency or not. Proposed method of Hurwitz-Radon

Matrices (MHR) is used in data extrapolation and then calculations for decision making are described. MHR method presents new approach to extrapolation problem because it takes the interpolation nodes to create orthogonal basis as columns of matrix OHR operators. Then affine (convex) combination of such basis builds new orthogonal base for unknown coordinates of calculated points. MHR method uses two-dimensional data for knowledge representation [5] and for computational foundations [6]. Also medicine [7], industry and manufacturing are looking for the methods connected with geometry of the curves [8]. So suitable data representation and precise reconstruction or extrapolation [9] of the curve is a key factor in many applications of computational intelligence, knowledge representation and computer vision.

2 Data Representation

Data are represented on the curve, described by the set of nodes $(x_i, y_i) \in \mathbf{R}^2$ (characteristic points) as follows in proposed method:

1. nodes (interpolation points) are settled at local extrema (maximum or minimum) of one of coordinates and at least one point between two successive local extrema;
2. nodes (x_i, y_i) are monotonic in coordinates x_i ($x_i < x_{i+1}$ for all i) or y_i ($y_i < y_{i+1}$);
3. one curve is represented by at least four nodes.

Condition 1 is done for the most appropriate description of a curve. The quotations or prices are real data for nodes. Condition 2 according to a graph of function means that coordinates x_i represent the time. Condition 3 is adequate for extrapolation, but in the case of interpolation minimal number of nodes is five.

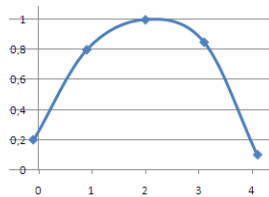


Fig. 1. Five nodes of data and a curve

Data points are treated as interpolation nodes. How can we extrapolate continues value in time for example $x = 4.1$ or discrete data for day $x = 5$ (Fig.1)? The anticipation of data is possible using novel MHR method.

3 Data Reconstruction

The following question is important in mathematics and computer sciences: is it possible to find a method of curve extrapolation in the plane without building the interpolation or extrapolation polynomials or other functions? Our paper aims at giving the positive answer to this question. In comparison MHR method with Bézier

curves, Hermite curves and B-curves (*B-splines*) or NURBS one unpleasant feature of these curves must be mentioned: small change of one characteristic point can make big change of whole reconstructed curve. Such a feature does not appear in MHR method. The methods of curve interpolation and extrapolation, based on classical polynomial interpolation: Newton, Lagrange or Hermite polynomials and the spline curves which are piecewise polynomials [10]. Classical methods are useless to interpolate the function that fails to be differentiable at one point, for example the absolute value function $f(x) = |x|$ at $x=0$. If point (0;0) is one of the interpolation nodes, then precise polynomial interpolation of the absolute value function is impossible. Also when the graph of interpolated function differs from the shape of polynomials considerably, for example $f(x) = 1/x$, interpolation and extrapolation is very hard because of existing local extrema of polynomial. Lagrange interpolation polynomial for function $f(x) = 1/x$ and nodes (5;0.2), (5/3;0.6), (1;1), (5/7;1.4), (5/9;1.8) has one minimum and two roots.

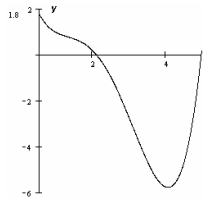


Fig. 2. Lagrange interpolation polynomial for nodes (5;0.2), (5/3;0.6), (1;1), (5/7;1.4), (5/9;1.8) differs extremely from the shape of function $f(x) = 1/x$

We cannot forget about the Runge’s phenomenon: when the interpolation nodes are equidistance then high-order polynomial oscillates toward the end of the interval, for example close to -1 and 1 with function $f(x) = 1/(1+25x^2)$ and extrapolation is impossible [11]. Method of Hurwitz – Radon Matrices (MHR), described in this paper, is free of these bad examples. The curve or function in MHR method is parameterized for value $\alpha \in [0;1]$ in the range of two successive interpolation nodes. MHR for data extrapolation is possible with $\alpha < 0$ or $\alpha > 1$.

3.1 The Operator of Hurwitz-Radon

Adolf Hurwitz (1859-1919) and Johann Radon (1887-1956) published the papers about specific class of matrices in 1923, working on the problem of quadratic forms. Matrices $A_i, i = 1,2...m$ satisfying

$$A_j A_k + A_k A_j = 0, A_j^2 = -I \text{ for } j \neq k; j, k = 1,2...m$$

are called a *family of Hurwitz - Radon matrices*. A family of Hurwitz - Radon (HR) matrices has important features [12]: HR matrices are skew-symmetric ($A_i^T = -A_i$) and reverse matrices are easy to find ($A_i^{-1} = -A_i$). Only for dimension $N = 2, 4$ or 8 the family of HR matrices consists of $N - 1$ matrices. For $N = 2$ we have one matrix:

$$A_1 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

For $N = 4$ there are three HR matrices with integer entries:

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}.$$

For $N = 8$ we have seven HR matrices with elements $0, \pm 1$.

So far HR matrices are applied in electronics [13]: in Space-Time Block Coding (STBC) and orthogonal design [14], also in signal processing [15] and Hamiltonian Neural Nets [16].

If one curve is described by a set of data points $\{(x_i, y_i), i = 1, 2, \dots, n\}$ monotonic in coordinates x_i (time for example), then HR matrices combined with the identity matrix I_N are used to build the orthogonal and discrete Hurwitz - Radon Operator (OHR). For nodes $(x_1, y_1), (x_2, y_2)$ OHR M of dimension $N = 2$ is constructed:

$$B = (x_1 \cdot I_2 + x_2 \cdot A_1)(y_1 \cdot I_2 - y_2 \cdot A_1) = \begin{bmatrix} x_1 & x_2 \\ -x_2 & x_1 \end{bmatrix} \begin{bmatrix} y_1 & -y_2 \\ y_2 & y_1 \end{bmatrix}, \quad M = \frac{1}{x_1^2 + x_2^2} B, \\ M = \frac{1}{x_1^2 + x_2^2} \begin{bmatrix} x_1 y_1 + x_2 y_2 & x_2 y_1 - x_1 y_2 \\ x_1 y_2 - x_2 y_1 & x_1 y_1 + x_2 y_2 \end{bmatrix}. \quad (1)$$

Matrix M in (1) is found as a solution of equation:

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}.$$

For nodes $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$, monotonic in x_i , OHR of dimension $N = 4$ is constructed:

$$M = \frac{1}{x_1^2 + x_2^2 + x_3^2 + x_4^2} \begin{bmatrix} u_0 & u_1 & u_2 & u_3 \\ -u_1 & u_0 & -u_3 & u_2 \\ -u_2 & u_3 & u_0 & -u_1 \\ -u_3 & -u_2 & u_1 & u_0 \end{bmatrix} \quad (2)$$

where

$$u_0 = x_1 y_1 + x_2 y_2 + x_3 y_3 + x_4 y_4, \quad u_1 = -x_1 y_2 + x_2 y_1 + x_3 y_4 - x_4 y_3, \\ u_2 = -x_1 y_3 - x_2 y_4 + x_3 y_1 + x_4 y_2, \quad u_3 = -x_1 y_4 + x_2 y_3 - x_3 y_2 + x_4 y_1.$$

Matrix M in (2) is found as a solution of equation:

$$\begin{bmatrix} a & b & c & d \\ -b & a & -d & c \\ -c & d & a & -b \\ -d & -c & b & a \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}.$$

For nodes $(x_1, y_1), (x_2, y_2), \dots, (x_8, y_8)$, monotonic in x_i , OHR of dimension $N = 8$ is built [17] similarly as (1) or (2):

$$M = \frac{1}{\sum_{i=1}^8 x_i^2} \begin{bmatrix} u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \\ -u_1 & u_0 & u_3 & -u_2 & u_5 & -u_4 & -u_7 & u_6 \\ -u_2 & -u_3 & u_0 & u_1 & u_6 & u_7 & -u_4 & -u_5 \\ -u_3 & u_2 & -u_1 & u_0 & u_7 & -u_6 & u_5 & -u_4 \\ -u_4 & -u_5 & -u_6 & -u_7 & u_0 & u_1 & u_2 & u_3 \\ -u_5 & u_4 & -u_7 & u_6 & -u_1 & u_0 & -u_3 & u_2 \\ -u_6 & u_7 & u_4 & -u_5 & -u_2 & u_3 & u_0 & -u_1 \\ -u_7 & -u_6 & u_5 & u_4 & -u_3 & -u_2 & u_1 & u_0 \end{bmatrix} \quad (3)$$

where

$$\mathbf{u} = \begin{bmatrix} \mathcal{Y}_1 & \mathcal{Y}_2 & \mathcal{Y}_3 & \mathcal{Y}_4 & \mathcal{Y}_5 & \mathcal{Y}_6 & \mathcal{Y}_7 & \mathcal{Y}_8 \\ -\mathcal{Y}_2 & \mathcal{Y}_1 & -\mathcal{Y}_4 & \mathcal{Y}_3 & -\mathcal{Y}_6 & \mathcal{Y}_5 & \mathcal{Y}_8 & -\mathcal{Y}_7 \\ -\mathcal{Y}_3 & \mathcal{Y}_4 & \mathcal{Y}_1 & -\mathcal{Y}_2 & -\mathcal{Y}_7 & -\mathcal{Y}_8 & \mathcal{Y}_5 & \mathcal{Y}_6 \\ -\mathcal{Y}_4 & -\mathcal{Y}_3 & \mathcal{Y}_2 & \mathcal{Y}_1 & -\mathcal{Y}_8 & \mathcal{Y}_7 & -\mathcal{Y}_6 & \mathcal{Y}_5 \\ -\mathcal{Y}_5 & \mathcal{Y}_6 & \mathcal{Y}_7 & \mathcal{Y}_8 & \mathcal{Y}_1 & -\mathcal{Y}_2 & -\mathcal{Y}_3 & -\mathcal{Y}_4 \\ -\mathcal{Y}_6 & -\mathcal{Y}_5 & \mathcal{Y}_8 & -\mathcal{Y}_7 & \mathcal{Y}_2 & \mathcal{Y}_1 & \mathcal{Y}_4 & -\mathcal{Y}_3 \\ -\mathcal{Y}_7 & -\mathcal{Y}_8 & -\mathcal{Y}_5 & \mathcal{Y}_6 & \mathcal{Y}_3 & -\mathcal{Y}_4 & \mathcal{Y}_1 & \mathcal{Y}_2 \\ -\mathcal{Y}_8 & \mathcal{Y}_7 & -\mathcal{Y}_6 & -\mathcal{Y}_5 & \mathcal{Y}_4 & \mathcal{Y}_3 & -\mathcal{Y}_2 & \mathcal{Y}_1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} \quad (4)$$

We can see here that the components of the vector $\mathbf{u} = (u_0, u_1, \dots, u_7)^T$, appearing in the matrix M (3), are defined by (4) in the similar way to (1)-(2) but in terms of the coordinates of the above 8 nodes. Note that OHR operators M (1)-(3) satisfy the condition of interpolation

$$M \cdot \mathbf{x} = \mathbf{y} \quad (5)$$

for $\mathbf{x} = (x_1, x_2, \dots, x_N)^T \in \mathbf{R}^N$, $\mathbf{x} \neq \mathbf{0}$, $\mathbf{y} = (y_1, y_2, \dots, y_N)^T \in \mathbf{R}^N$, $N = 2, 4$ or 8 .

If one curve is described by a set of nodes $\{(x_i, y_i), i = 1, 2, \dots, n\}$ monotonic in coordinates y_i , then HR matrices combined with the identity matrix I_N are used to build the orthogonal and discrete reverse Hurwitz - Radon Operator (reverse OHR) M^{-1} . If matrix M in (1)-(3) is described as:

$$M = \frac{1}{\sum_{i=1}^N x_i^2} (u_0 \cdot I_N + D),$$

where matrix D consists of elements 0 (diagonal), u_1, \dots, u_{N-1} , then reverse OHR M^{-1} is given by:

$$M^{-1} = \frac{1}{\sum_{i=1}^N y_i^2} (u_0 \cdot I_N - D). \quad (6)$$

Note that reverse OHR operator (6) satisfies the condition of interpolation

$$M^{-1} \cdot \mathbf{y} = \mathbf{x} \quad (7)$$

for $\mathbf{x} = (x_1, x_2, \dots, x_N)^T \in \mathbf{R}^N$, $\mathbf{y} = (y_1, y_2, \dots, y_N)^T \in \mathbf{R}^N$, $\mathbf{y} \neq \mathbf{0}$, $N = 2, 4$ or 8 .

3.2 MHR Method and Data Extrapolation

Key question looks as follows: how can we compute coordinates of points settled between the interpolation nodes [18] or beyond the nodes? The answer is connected with MHR method for interpolation [19] and extrapolation. On a segment of a line every number “ c ” situated between “ a ” and “ b ” is described by a linear (convex) combination $c = \alpha \cdot a + (1 - \alpha) \cdot b$ for

$$\alpha = \frac{b - c}{b - a} \in [0;1]. \tag{8}$$

If $c < a$ then $\alpha > 1$: extrapolation of points situated left of nodes. If $c > b$ then $\alpha < 0$: extrapolation of points situated right of nodes.

When the nodes are monotonic in coordinates x_i , the average OHR operator M_2 of dimension $N = 2, 4$ or 8 is constructed as follows:

$$M_2 = \alpha \cdot M_0 + (1 - \alpha) \cdot M_1 \tag{9}$$

with the operator M_0 built (1)-(3) by “odd” nodes $(x_1=a, y_1), (x_3, y_3), \dots, (x_{2N-1}, y_{2N-1})$ and M_1 built (1)-(3) by “even” nodes $(x_2=b, y_2), (x_4, y_4), \dots, (x_{2N}, y_{2N})$. Having the operator M_2 for coordinates $x_i < x_{i+1}$ it is possible to reconstruct the second coordinates of points (x, y) in terms of the vector C defined with

$$c_i = \alpha \cdot x_{2i-1} + (1 - \alpha) \cdot x_{2i} \quad , \quad i = 1, 2, \dots, N \tag{10}$$

as $C = [c_1, c_2, \dots, c_N]^T$. The required formula is similar to (5):

$$Y(C) = M_2 \cdot C \tag{11}$$

in which components of vector $Y(C)$ give the second coordinate of the points (x, y) corresponding to the first coordinate, given in terms of components of the vector C .

On the other hand, having the operator M_2^{-1} for coordinates $y_i < y_{i+1}$ it is possible to reconstruct the first coordinates of points (x, y) :

$$M_2^{-1} = \alpha \cdot M_0^{-1} + (1 - \alpha) \cdot M_1^{-1}, \quad c_i = \alpha \cdot y_{2i-1} + (1 - \alpha) \cdot y_{2i}, \tag{12}$$

$$X(C) = M_2^{-1} \cdot C \quad .$$

Calculation of unknown coordinates for curve points using (8)-(12) is called by author the method of Hurwitz - Radon Matrices (MHR) [20]. Here is the application of MHR method for functions $f(x) = 1/x$ (nodes as Fig. 2) and $f(x) = 1/(1+25x^2)$ with five nodes equidistance in first coordinate: $x_i = -1, -0.5, 0, 0.5, 1$.

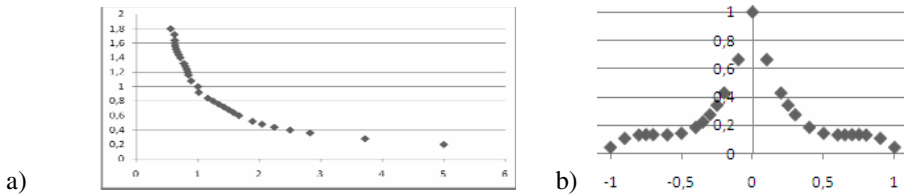


Fig. 3. Twenty six interpolated points of functions $f(x)=1/x$ (a) and $f(x) = 1/(1+25x^2)$ (b) using MHR method with 5 nodes

MHR interpolation for function $f(x) = 1/x$ gives better result than Lagrange interpolation (Fig. 2). The same can be said for function $f(x) = 1/(1+25x^2)$ [21].

MHR extrapolation is valid for $\alpha < 0$ or $\alpha > 1$. In the case of continuous data, parameter α is a real number. For example there are four nodes: (1;2), (1.3;5), (2;3), (2.5;6). MHR extrapolation with $\alpha = -0.01$ gives the point (2.505;6.034) and with $\alpha = -0.1$: (2.55;6.348). But the rate of a currency or the quotations are discrete data. If we assume that the rate of a currency is represented by equidistance nodes (day by day – fixed step of time $h = 1$ for coordinate x), next data or the rate on next day is extrapolated (anticipated) for $\alpha = -1$.

4 Decision Making

Here are MHR calculations for true rates of Euro at National Bank of Poland (NBP) from January 24th to February 14th, 2011. When the last four rates are being considered: (1;3.8993), (2;3.9248), (3;3.9370) and (4;3.9337), MHR extrapolation with matrices of dimension $N = 2$ gives the result (5;3.9158). So anticipated rate of Euro on the day February 15th is 3.9158.

If the last eight rates are being considered: (1;3.9173), (2;3.9075), (3;3.8684), (4;3.8742), (5;3.8993), (6;3.9248), (7;3.9370) and (8;3.9337), MHR extrapolation with matrices of dimension $N = 4$ gives the result (9;4.0767). Anticipated rate of Euro on the day February 15th is 4.0767. There are two extrapolated values for next day. This example gives us two anticipated rates for tomorrow: 3.9158 and 4.0767. How these extrapolated values can be used in the process of decision making: to buy Euro or not, to sell Euro or not? The proposal final anticipated rate of Euro for the day February 15th based on weighted mean value:

$$\frac{2 \cdot 3.9158 + 4.0767}{3} = 3.9694 \quad (13)$$

because the rate 3.9158 is calculated for $N = 2$, whereas 4.0767 is extrapolated for $N = 4$.

If the last sixteen rates are being considered, MHR extrapolation with matrices of dimension $N = 8$ has to be used. Here are the rates: (1;3.8765), (2;3.8777), (3;3.8777), (4;3.9009), (5;3.9111), (6;3.9345), (7;3.9129), (8;3.9019), (9;3.9173), (10;3.9075), (11;3.8684), (12;3.8742), (13;3.8993), (14;3.9248), (15;3.9370) and (16;3.9337). Average OHR operator M_2 and MHR calculations look as follows:

$$M_2 = \begin{bmatrix} 0.3226 & 0.0154 & 0.0286 & 0.0462 & -0.062 & 0.0444 & 0.0924 & 0.0461 \\ -0.0154 & 0.3226 & 0.0462 & -0.0286 & 0.0444 & 0.062 & -0.0461 & 0.0924 \\ -0.0286 & -0.0462 & 0.3226 & 0.0154 & 0.0924 & 0.0461 & 0.062 & -0.0444 \\ -0.0462 & 0.0286 & -0.0154 & 0.3226 & 0.0461 & -0.0924 & 0.0444 & 0.062 \\ 0.062 & -0.0444 & -0.0924 & -0.0461 & 0.3226 & 0.0154 & 0.0286 & 0.0462 \\ -0.0444 & -0.062 & -0.0461 & 0.0924 & -0.0154 & 0.3226 & -0.0462 & 0.0286 \\ -0.0924 & 0.0461 & -0.062 & -0.0444 & -0.0286 & 0.0462 & 0.3226 & -0.0154 \\ -0.0461 & -0.0924 & 0.0444 & -0.062 & -0.0462 & -0.0286 & 0.0154 & 0.3226 \end{bmatrix}, \quad M_2 \begin{bmatrix} 3 \\ 5 \\ 7 \\ 9 \\ 11 \\ 13 \\ 15 \\ 17 \end{bmatrix} = \begin{bmatrix} 3.7252 \\ 3.8072 \\ 3.8704 \\ 3.8278 \\ 3.8653 \\ 3.8834 \\ 3.9825 \\ 3.9882 \end{bmatrix}$$

MHR extrapolation gives the result (17;3.9882). Anticipated rate of Euro on the day February 15th is 3.9882.

MHR extrapolation has been done for three times ($N = 2, 4, 8$) and anticipated values are 3.9158, 4.0767 and 3.9882 respectively. The proposal final anticipated rate of Euro for the day February 15th based on weighted mean value:

$$\frac{4 \cdot 3.9158 + 2 \cdot 4.0767 + 3.9882}{7} = 3.9721 \quad (14)$$

because the rate 3.9158 is calculated with last four data points, 4.0767 is extrapolated for last eight data points and 3.9882 is computed for last sixteen data points.

The true rate of Euro for the day February 15th is 3.9398. In author's opinion, extrapolated rates 3.9694 (13) and 3.9721 (14) for next day preserve the increasing trend and they are good enough to be one of the factors for making a decision of buying or selling the currency. Anticipated values, calculated by MHR method, are applied in the process of decision making: to follow the action or not, to do one thing or another. Extrapolated values can be used to make a decision in many branches of science and economics.

5 Conclusions

The method of Hurwitz-Radon Matrices leads to curve interpolation and value extrapolation depending on the number and location of data points. Proposed MHR method uses the orthogonal basis created by interpolation nodes and orthogonal OHR matrix operator is applied to interpolation and extrapolation. No characteristic features of curve are important in MHR method: failing to be differentiable at any point, the Runge's phenomenon or differences from the shape of polynomials. These features are very significant for classical polynomial interpolations and extrapolations. MHR method gives the possibility of reconstruction a curve and anticipation the data points. The only condition is to have a set of nodes according to assumptions in MHR method. Data representation and curve extrapolation by MHR method is connected with possibility of changing the nodes coordinates and reconstruction of new data or curve for new set of nodes. The same MHR interpolation and extrapolation is valid for discrete and continues data. Main features of MHR method are: accuracy of data reconstruction depending on number of nodes; interpolation or extrapolation of a curve consists of L points is connected with the computational cost of rank $O(L)$; MHR method is dealing with local operators: average OHR operators are built by successive 4, 8 or 16 data points, what is connected with smaller computational costs then using all nodes; MHR is not an affine interpolation [22].

Future works are connected with: MHR application in parallel processing [23] (curve interpolations and extrapolations that can be computed for two or more curves with data independencies), implementation in sequential processes [24], possibility to apply MHR method to three-dimensional curves (3D data), computing the extrapolation error, object recognition [25] and MHR version for equidistance nodes.

References

1. Brachman, R.J., Levesque, H.J.: Knowledge Representation and Reasoning. Morgan Kaufman, San Francisco (2004)
2. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning About Knowledge. MIT Press, Cambridge (1995)
3. Straffin, P.D.: Game Theory and Strategy. Mathematical Association of America, Washington, D.C (1993)
4. Watson, J.: Strategy – An Introduction to Game Theory. University of California, San Diego (2002)
5. Markman, A.B.: Knowledge Representation. Lawrence Erlbaum Associates, Mahwah (1998)
6. Sowa, J.F.: Knowledge Representation: Logical, Philosophical and Computational Foundations. Brooks/Cole, New York (2000)
7. Soussen, C., Mohammad-Djafari, A.: Polygonal and Polyhedral Contour Reconstruction in Computed Tomography. IEEE Transactions on Image Processing 11(13), 1507–1523 (2004)
8. Tang, K.: Geometric Optimization Algorithms in Manufacturing. Computer – Aided Design & Applications 2(6), 747–757 (2005)
9. Kozera, R.: Curve Modeling via Interpolation Based on Multidimensional Reduced Data. Silesian University of Technology Press, Gliwice (2004)
10. Dahlquist, G., Björck, A.: Numerical Methods. Prentice Hall, New York (1974)
11. Ralston, A.: A First Course in Numerical Analysis. McGraw-Hill Book Company, New York (1965)
12. Eckmann, B.: Topology, Algebra, Analysis- Relations and Missing Links. Notices of the American Mathematical Society 5(46), 520–527 (1999)
13. Citko, W., Jakóbczak, D., Sieńko, W.: On Hurwitz - Radon Matrices Based Signal Processing. In: Workshop Signal Processing at Poznan University of Technology (2005)
14. Tarokh, V., Jafarkhani, H., Calderbank, R.: Space-Time Block Codes from Orthogonal Designs. IEEE Transactions on Information Theory 5(45), 1456–1467 (1999)
15. Sieńko, W., Citko, W., Wilamowski, B.: Hamiltonian Neural Nets as a Universal Signal Processor. In: 28th Annual Conference of the IEEE Industrial Electronics Society IECON (2002)
16. Sieńko, W., Citko, W.: Hamiltonian Neural Net Based Signal Processing. In: The International Conference on Signal and Electronic System ICSES (2002)
17. Jakóbczak, D.: 2D and 3D Image Modeling Using Hurwitz-Radon Matrices. Polish Journal of Environmental Studies 4A(16), 104–107 (2007)
18. Jakóbczak, D.: Shape Representation and Shape Coefficients via Method of Hurwitz-Radon Matrices. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2010. LNCS, vol. 6374, pp. 411–419. Springer, Heidelberg (2010)
19. Jakóbczak, D.: Curve Interpolation Using Hurwitz-Radon Matrices. Polish Journal of Environmental Studies 3B(18), 126–130 (2009)
20. Jakóbczak, D.: Application of Hurwitz-Radon Matrices in Shape Representation. In: Banaszak, Z., Świć, A. (eds.) Applied Computer Science: Modelling of Production Processes, vol. 1(6), pp. 63–74. Lublin University of Technology Press, Lublin (2010)
21. Jakóbczak, D.: Object Modeling Using Method of Hurwitz-Radon Matrices of Rank k . In: Wolski, W., Borawski, M. (eds.) Computer Graphics: Selected Issues, pp. 79–90. University of Szczecin Press, Szczecin (2010)

22. Jakóbczak, D.: Implementation of Hurwitz-Radon Matrices in Shape Representation. In: Choraś, R.S. (ed.) Image Processing and Communications Challenges 2. Advances in Intelligent and Soft Computing, vol. 84, pp. 39–50. Springer, Heidelberg (2010)
23. Chang, F.-C., Huang, H.-C.: A Refactoring Method for Cache-Efficient Swarm Intelligence Algorithms. Information Sciences (2010), doi:10.1016/j.ins.2010.02.025

The Memetic Ant Colony Optimization with Directional Derivatives Simplex Algorithm for Time Delays Identification

Janusz P. Papliński

Department of Control and Measurement, West Pomeranian University of Technology,
26 Kwietnia 10, 71-126 Szczecin, Poland
janusz.paplinski@zut.edu.pl

Abstract. The identification of time delay in the linear plant is important tasks. Most of the conventional identification techniques, such as those based on least mean-squares, are essentially gradient-guided local search techniques and they require a smooth search space or a differentiable performance index. New possibility in this field is opened by an application of the hybrid Ant Colony Optimization (ACO) with local optimization algorithm. The Directional Derivatives Simplex (DDS) as a local optimization algorithm is proposed in the paper and used in the memetic ACODDS method. The ACODDS algorithm is compared with ACO and a classical methods: Global Separable Nonlinear Least Squares (GSNLS). The obtained results suggest that the proposed method performs well in estimating the model parameters.

Keywords: Ant Colony Optimization, Nelder–Mead simplex, time delay, identification, memetic algorithm.

1 Introduction

Knowledge of the transport delays, that often occurs in linear objects, is very important when the control system is designed. If parameters used for controller design does not coincide with the real process time delays, then the close-loop system can be unstable or may cause efficiency lost [1, 2, 3]. The identification of time delays in linear systems is important and should be treated as the first task during system analysis and control design. This problem can become more complicated for the multi-input single-output system (MISO), where the solution space is multimodal.

Most of the conventional system identification techniques, such as those based on the non-linear estimation method, for example Separable Nonlinear Least Squares Method (SEPNLS), are in essence gradient-guided local search methods [4]. They require a smooth search space or a differentiable performance index. The conventional approaches in the multi-modal optimisation can easily fail in obtaining the global optimum and may be stopped at local optimum [5, 6]. One of the possible solution of this problem is use of a SEPNLS methods with global optimisation elements (GSNLS) [4, 7]. New possibility of identification of systems with multi modal solution space was opened by application of the computational intelligence

methods [8, 9, 10, 11, 12]. Ant Colony Optimization (ACO) is one among them. Ants are known as a social insects. They exhibit adaptive and flexible collective behavior to achieve various tasks. The macro-scale complex behavior emerges as a result of cooperation in micro-scale [13].

In order to improve the power of optimal solution search the hybrid schemes are proposed. A memetic Ant Colony Optimization algorithm (ACO) as a collective computation algorithm, with a modified Nelder-Mead simplex (NM) as a local search method, is presented in the paper.

This paper considers the problem of parameter estimation from sampled input-output data for continuous-time systems with unknown time delays. The time delay of the plant is identified by one of the proposed in the paper algorithms and streamline by using memetic Ant Colony with Directional Derivatives Simplex algorithm (ACODDS). The Global Separable Nonlinear Least-Squares (GSNLS) method [14] are also presented in the paper as a comparative methods. The linear parameters of the model is obtained by using linear LS method.

2 Problem Description

The dynamic of continuous-time MISO system with unknown time delays can be described as:

$$\sum_{i=0}^n a_i p^{n-i} x(t) = \sum_{j=1}^r \sum_{k=1}^{m_j} b_{jk} p^{m_j-k} u_j(t - t_{dj}), \quad (1)$$

where $a_0 = 1$, $b_{j1} \neq 0$, p – differential operator, $u_j(t)$ – j -th input, t_{dj} – time delay of j -th input, x – non-disturbed output of system. We assume the parameters n and m_j are known. The measurement output is disturbed by stochastic noise:

$$y(t) = x(t) + v(t). \quad (2)$$

The zero order hold, with sampling period T , is used

$$u_j(t) = \tilde{u}_j(k). \text{ for } (k-1)T \leq t < kT, \quad (3)$$

The problem studied here is as follows: how to estimate the time delays and the system parameters from sampled data representation of the inputs and the noisy output.

3 The Optimisation Algorithms

3.1 Ant Colony Optimization (ACO)

Researchers in various fields have showed interest in the behaviour of social creatures to solve various tasks [15, 16]. The ants exhibit collective behaviour to perform tasks as foraging, building a nests. These tasks can not be carried out by one individual. The macro-scale complex behaviour emerges as a result of cooperation in micro-scale. This appears without any central or hierarchical control. A way of

communicating between individuals in colony is chemical substances called pheromones [17]. Ants looking for food lay, the way back to their nest, a specific type of pheromone. Other ants can follow the pheromone trail and find the way to the aliments. Pheromones remain in the some way superimpose and intensify, concurrently the pheromones evaporate in time and their intensity decreases. A specific map of pheromones is created on the search space. This map is not immutable and during the iteration will adapt to the environment. Each ant looks for food independently of the others and moves from nest to source of food. There are a lot of ways in which ants can go. Ants choose a path using three sources of information:

- the own experience
- the local information
- the pheromone trail.

The own experience permits to recognize place when it already was and avoid looping of the way. For the artificial ant it permits to allocate particular value to appropriate seeking parameters. The local information determines permissible way. Ants can recognize and sidestep hindrances. In the artificial ant colony it is responsible for the search space constraints. The pheromone trail permits to come back to the nest and find the source of food found earlier by another individuals from colony. Ants prefer this way in which intensity of pheromones is the biggest.

Ants leaves pheromones trace which intensity $\tau_i(t)$ is given by the equation:

$$\tau_i(t) = \frac{n}{m} \frac{J_i^\alpha}{\sum_{k=1}^n J_k^\alpha} . \tag{4}$$

where: J_i – a quality function of solution find by i -th ant, n – an amount of ant in the nest, m – an amount of pheromone trace, α – a parameter that control the exploration/exploitation mechanism by influence on the ratio of the pheromone trace leaved by the best and the worst ant. The quality function J is divided by the sum of all quality functions in order to uniform it to one. The ratio of the number of ants to the number of rows in the matrix of pheromones, scales intensity of leaving new pheromones to the already existing traces. At every iteration the existing pheromone evaporate

$$\tau_i(t+1) = \rho \tau_i(t) . \tag{5}$$

where ρ is a glow of pheromone represents the evaporation rate. The value of ρ must be set less than 1 to avoid unlimited accumulation of the pheromone. The best results are obtained for $\rho = 0,925$. The amount of pheromone traces was limited to specified number $m = \beta n$, by removing the worst traces in each iteration. The balance between exploration and extrapolation during optimization is controlled by evaporation rate ρ and tunable parameter β .

The time delays of the model (1) are interpreted as a decision node inside a way of ants. It is a place where an ant has to decide about next direction of motion. The discrete probability distribution function, used to generate new candidate solution from existing trace at iteration t , is defined as:

$$p_{ij}(t) = \frac{\tau_{ij}(t)}{\sum_{k=1}^m \tau_k(t)}, \quad (6)$$

where: i – the sequential number of pheromone trace, j – the number of parameter. This function has discrete domain when the time delay can take a real value. The transition from discrete probability distribution function to continuous one is obtained by applying, between existing points in the pheromone trace, linear approximation. It is different method than proposed by Socha and Dorigo [18].

The half of population of ants has disturbed direction additional:

$$t_{dij} = \zeta_{ijn}\beta + \zeta_{ijs}(1 - \beta), \quad (7)$$

where: i – the number of ants, j – the number of parameter, ζ_{ijn} – the random value with normal distribution, inside the solution space, ζ_{ijs} – the random value with distribution defined by linear approximation and (6), β – a random coefficient of ratio of averaging.

The ants are looking only for the time delays of a model. The residual parameters of the model are obtained by SEP-NLS during calculation the quality function of individuals. It can be do because these parameters are linear and SEP-NLS works efficiently with them. The SEP-NLS algorithm is described below.

3.2 The Directional Derivatives Simplex (DDS)

The simplex search method was proposed by Spendley, Hext and Himsforth [19], and next redefined by Nelder Mead [20]. It is a derivative-free optimization technique based on the comparison among the cost function values at the $n + 1$ vertices of the simplex (polytope). The value of n is equal to the dimension of the search space. The Nelder-Mead method (NM) has been developed and allied optimization techniques was created [21, 22]. The Directional Derivatives Simplex (DDS) is one of this methods. The DDS changes the position of centroid, relative to NM, shifting him towards the best vertex. The new defined centroid x_c can be determined as weighted sum by using the directional derivatives as a weight coefficient:

$$x_c = \frac{\sum_{i=2}^{n+1} f'_{x_1 x_i}(x_i) x_i}{\sum_{i=2}^{n+1} f'_{x_1 x_i}(x_i)}, \quad (8)$$

where the directional derivative $f'_{\overline{x_1x_i}}(x_i)$ is defined as:

$$f'_{\overline{x_1x_i}}(x_i) = \frac{f(x_i) - f(x_1)}{|\overline{x_1x_i}|} = \frac{f(x_1 + \overline{x_1x_i}) - f(x_1)}{|\overline{x_1x_i}|}, \tag{9}$$

where $|\overline{x_1x_i}|$ is the length of vector $\overline{x_1x_i}$ connecting the worst vertex x_1 with the one of the other vertex x_i . The vertexes are sorted from the worst x_1 to the best x_{n+1} :

$$f(x_1) \geq f(x_2) \geq \dots \geq f(x_{n+1}). \tag{10}$$

The vertex with the lowest value of function is replaced by a newly reflected, better point:

$$x_n = (1 + \alpha)x_c + \alpha x_1. \tag{11}$$

The reflected point can be obtained by using the same procedure as in the classical NM algorithm. In this case the coefficient α takes one of the following values:

$$\alpha = \{1, 2, 0.5, -0.5\}. \tag{12}$$

3.3 The Memetic Ant Colony Optimisation with Directional Derivatives Simplex Algorithm (ACODDS)

The memetic algorithms (MA) is a combination of any population based approach, and a local search method [23]. The concept meme has been defined by Dawkins [24], and the term Memetic Algorithm was coined by Moscato [25]. MAs have demonstrated better results in a great variety of problems [26, 27]. In this class of hybrid algorithms, the local search operator is used to improve individuals of the population during the computation cycle.

Application the DDS into ACODDS permits to improve the solution and leave better trace on the pheromone trail by selected ant in every iteration. The best three individuals into ant population are selected in each iteration and simplex is created on the based of their. The centroid is computed by using equation (7). A newly reflected ant is obtained by using equation (10) with random coefficient α . The random variable α has the uniform distribution:

$$\alpha \sim U(-0.5; 0.5). \tag{13}$$

This individual leaves its mark on the pheromone trail which is better: the new one created by DDS or the worst of the selected to the simplex in DDS. The applied mechanism allows to improve some individuals based on the two best members of the population.

3.4 The Global Separable Nonlinear Least Squares (GSNLS) Estimation Method

The linear parameters of the model can be estimated by using SEP-NLS, as the minimizing arguments of the LS criterion [8]:

$$V_N(\theta, t_d) = \frac{1}{N - k_s} \sum_{k=ks+1}^N \frac{1}{2} \varepsilon^2(k\theta, t_d) = \frac{1}{N - k_s} \sum_{k=ks+1}^N \frac{1}{2} \left(y(t) - \varphi^T(k, t_d) \theta \right)^2. \quad (14)$$

The vectors of the time delays t_d and linear parameters θ are estimated in a separable manner. The linear parameters, when the time delays are known, can be obtained from linear LS method:

$$\theta = \arg \min_{\theta} V_N(\theta, t_d). \quad (15)$$

The time delays t_d can be estimated as the minimizing arguments of the criterion

$$\hat{t}_d = \arg \min_{t_d} \tilde{V}_N(t_d). \quad (16)$$

The SEP-NLS method can converge to the local optimum. It is possible to apply stochastic approximation [28] with convolution smoothing to the SEP-NLS method in order to reach the global optimum [29]. The estimate of the time delay in GSNLS can be obtained as follows:

$$\hat{t}_{d_j}^{(l+1)} = \hat{t}_{d_j}^{(l)} - \mu^{(l)} \left(\left(\tilde{R}_j \left(\hat{t}_{d_j}^{(l)} \right) \right)^{-1} \tilde{V}_j' \left(\hat{t}_{d_j}^{(l)} \right) + \beta^{(l)} \eta \right), \quad (17)$$

where β is a random value disturbing time delay obtained by SEP-NLS.

4 Experimental Results

The algorithm presented in the previous sections has been tested on the MISO example. We consider the following system [30]:

$$\ddot{y}(t) + a_1 \dot{y}(t) + a_2 y(t) = b_{11} u_1(t - t_{d1}) + b_{21} u_2(t - t_{d2}), \quad (18)$$

where $a_1=3.0$; $a_2=4.0$; $b_{11}=2.0$; $b_{21}=2.0$; $t_{d1}=9.15$; $t_{d2}=2.57$. The inputs and output signals are converted by zero-order-hold operation with sampling period $T=0.05$. The pre-filter $Q(p)$ with $\alpha=0.4$ is used. As an input signal an independent sequence of uniform distribution between 0 and 1 is used. The signal to measurement noise ratio SNR is 5%. A data set of 1000 samples was generated for the identification process. The algorithms are implemented for 250 iterations. The initial values of time delays $t_d^{(0)}$ are randomly chosen between 0 and 25. All algorithms, presented in the paper were running 100 times. The GSNLS was used with variance coefficient $\mu=10^6$.

The solution space of time delays is multimodal [31] and the global optimum is not reached in every time. The percentages of identified time delays, with error less than 10% $\delta_{t_{d1}}$, $\delta_{t_{d2}}$, can be treated as a criterion of algorithm convergence and they are presented in the Table 1. For the GSNLS the global optimum are reached only at 84%

of trials. The ACO and ACODDS work well and in the experiments they always achieved the global optimum. The main goal of identification is finding the unknown input delays of the plant. The obtained average value of time delay with standard deviation are also presented in the Table 1. The worst results were obtained for the GSNLS method, as was expected. Especially for the time t_{d2} identification error and standard deviation are very large, this results from the difficulties in finding a global optimum. The ACO and ACODDS give similar results with a slight predominance of the memetic method ,especially for identification of time delay t_{d1} .

All methods give comparable results if we take into account just correct identification with error less than 10%. It is shown in the Table 2. The GSNLS method is only little worse then another methods. The performance of identification algorithms is determined by accuracy and time of computing. The number of functional evaluations required to reach true time delay with 10% accuracy is presented in Fig. 1. The average time of computing is about 5% smaller for ACODDS algorithm than for ACO, and about 64% smaller than GSNLS needs.

Table 1. The percent of identified time delay $\delta_{t_{d1}}$, $\delta_{t_{d2}}$ which are reached the global optimum, and the average value of time delay with standard deviation for identification of the plant with input delays equal $t_{d1}=9.15$; $t_{d2}=2.57$

	$\delta_{t_{d1}}$ %	$\delta_{t_{d2}}$ %	avg. t_{d1}	std dev. t_{d1} %	avg. t_{d2}	std dev. t_{d2} %
GSNLS	100%	84%	9,32	14%	5,32	256%
ACO	100%	100%	9,01	4%	2,57	6%
ACODDS	100%	100%	9,12	3%	2,59	6%

Table 2. The average value of time delay with standard deviation, just for the correct identification, with error less than 10%

	avg. t_{d1}	std dev. t_{d1} %	avg. t_{d2}	std dev. t_{d2} %
GSNLS	9,16	2%	2,55	6%
ACO	9,01	4%	2,57	6%
ACODDS	9,12	3%	2,59	6%

The median value of computation time is smaller then average. This indicates that in a series of 100 trials were a few strongly worse. The advantage of ACODDS in this case is much more visible. The median value of running time for ACODDS is about 23% smaller than for ACO, and about 58% smaller than for GSNLS.

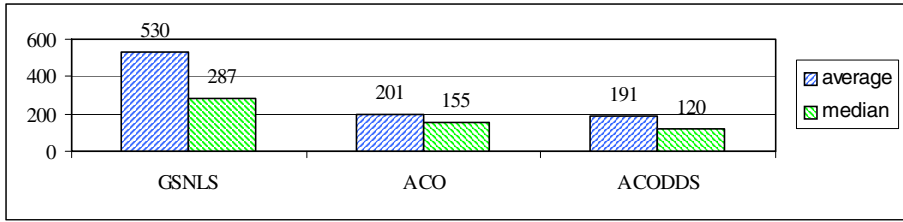


Fig. 1. The average and median number of calls of quality function required to reach true time delay with 10% accuracy

5 Conclusion

Three algorithms used for identification of systems with time delays are presented in the paper: the classical algorithm GSNLS which is a global version of the SEP-NLS algorithm, the computational collective algorithm ACO, and the memetic algorithm ACODDS.

The worst method, in terms of both quality and speed, was the GSNLS. The algorithm ACO is method that allows to significantly improve the quality of the identified delay and reduce the time needed for perform calculations. The pheromone trace mechanism was an effective tool for multimodal optimization.

The hybrid algorithm ACODDS, that uses modified NM to improve the solutions obtained in a single iteration by ACO, allows for better performance of optimizations. The calculation time is reduced especially, as it can be seen on the median values.

A directional derivative include in NM (DDS) allows to move direction of exploration a new solutions towards the best vertex. This allows to achieve a better solutions and improves properties of algorithm.

References

1. Bjorklund, S., Ljung, L.: A Review of Time-Delay Estimation Techniques. In: Proceedings of the IEEE Conference on Decision and Control 2003, Maui, Hawaii, USA, vol. 3, pp. 2502–2507 (2003)
2. Boukas, E.K.: Stochastic output feedback of uncertain time-delay system with saturating actuators. *Journal of optimization theory and applications* 118(2), 255–273 (2003)
3. Li, Z.-S., Wan, Y.-C.: Suboptimal control for plants with pure time delay based on state feedback. *Shanghai Jiaotong Daxue Xuebao/Journal of Shanghai Jiaotong University* 36(suppl.), 138–140 (2002)
4. Chen, X., Wang, M.: Global optimization methods for time delay estimation. In: Proceedings of the World Congress on Intelligent Control and Automation (WCICA), vol. 1, pp. 212–215 (2004)
5. Chen, B.-S., Hung, J.-C.: A global estimation for multichannel time-delay and signal parameters via genetic algorithm. *Signal Processing* 81(5), 1061–1067 (2001)
6. Harada, K., Kobayashi, Y., Okita, T.: Identification of Linear Systems With Time Delay and Unknown Order Electrical. *Engineering in Japan (English translation of Denki Gakkai Ronbunshi)* 145(3), 61–68 (2003)

7. Previdi, F., Lovera, M.: Identification of non-linear parametrically varying models using separable least squares. *Int. J. Control* 77(16), 1382–1392 (2004)
8. Olinsky, A.D., Quinn, J.T., Mangiameli, P.M., Chen, S.K.: A genetic algorithm approach to nonlinear least squares estimation. *Int. J. Math. Educ. SCI. Technol.* 35(2), 207–217 (2004)
9. Papiński, J.P.: An evolutionary algorithm for identification of non-stationary linear plants with time delay. In: *Proceedings of the First International Conference of Informatics in Control, Automation and Robotics (ICINCO)*, vol. 1, pp. 64–69 (2004)
10. Phat, V.N., Savkin, A.V.: Robust state estimation for a class of uncertain time-delay systems. *Systems and Control Letters* 47(3), 237–245 (2002)
11. Shaltaf, S.: Neural-Network-Based Time-Delay Estimation. *Eurasip. Journal on Applied Signal Processing* 3, 378–385 (2004)
12. Yang, Z., Iemura, H., Kanae, S., Wada, K.: A global nonlinear instrumental variable method for identification of continuous-time systems with unknown time delays. *IFAC World Congress, Prague, Czech Republic* (2005)
13. Bonabeau, E., Dorigo, M., Theraulaz, G.: *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York (1999)
14. Westwick, D.T., Kearney, R.E.: Separable least squares identification of nonlinear Hammerstein models: application to stretch reflex dynamics. *Annals of Biomedical Engineering* 29, 707–718 (2001)
15. Dorigo, M., Stützle, T.: *Ant colony optimization*. MIT Press, Cambridge (2004)
16. Andries, P.: *Engelbrecht. Fundamentals of Computational Swarm Intelligence*. Wiley and Sons, Ltd., England (2006)
17. Agosta, W.C.: *Chemical communication – the Language of Pheromone*. W.H. Freeman and Company, New York (1992)
18. Socha, K., Dorigo, M.: Ant colony optimization for continuous domains. *European Journal of Operational Research* 185, 1155–1173 (2008)
19. Spendley, W., Hext, G.R., Himsworth, F.R.: Sequential application of simplex design in optimization and evolutionary operation. *Technometrics* 4, 441–461 (1962)
20. Nelder, J.A., Mead, R.: A simplex method for function minimization. *The Computer Journal* 7, 308–313 (1965)
21. Ryan, P.B., Barr, R.L., Tod, H.D.: Simplex Techniques for Nonlinear Optimization. *Anal. Chem.* 52(9), 1460–1467 (1980)
22. Umeda, T., Kawa, A.I.: A Modified Complex Method for Optimization. *Ind. Eng. Chem. Process Des. Develop.* 10(2), 229–236 (1971)
23. Glover, F., Kochenberger, G.A. (eds.): *Handbook of Metaheuristics*. International Series in Operations Research & Management Science, vol. 57. Kluwer Academic Publishers/Springer, New York, USA (2003)
24. Moscato, P.: Memetic algorithms. In: *Handbook of Applied Optimization*, ch. 3.6.4, pp. 157–167. Oxford University Press, Oxford (2002)
25. Dawkins, R.: *The Selfish Gene*, 1st edn. Oxford University Press, Oxford (1976), 2 edn. (October 1989)
26. Mavrovouniotis, M., Yang, S.: A memetic ant colony optimization algorithm for the dynamic traveling salesman problem. In: *Soft Computing - A Fusion of Foundations, Methodologies and Applications* (2010)
27. Wang, F., Qiu, Y.-h.: Multimodal Function Optimizing by a New Hybrid Nonlinear Simplex Search and Particle Swarm Algorithm. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) *ECML 2005. LNCS (LNAI)*, vol. 3720, pp. 759–766. Springer, Heidelberg (2005)

28. Bharth, B., Borkar, V.S.: Stochastic approximation algorithms: overview and recent trends. *Sādhanā India* 24, Parts 4 & 5, 425–452 (1999)
29. Iemura, H., Yang, Z., Kanae, S., Wada, K.: Identification of continuous-time systems with unknown time delays by global nonlinear least-squares method. In: *IFAC Workshop on Adaptation and Learning in Control and Signal Processing*, Yokohama, Japan (2004)
30. Papliński, J.P.: Hybrid genetic and Nelder-Mead algorithms for identification of time delay. In: *Proceedings of the 14th IEEE IFAC International Conference Methods and Models in Automation and Robotics (MMAR)*, Międzyzdroje, Poland (2009)
31. Papliński, J.P.: The genetic algorithm with simplex crossover for identification of time delay. In: *Intelligent Information Systems, New Approaches*, pp. 337–346. Publishing House of University of Podlasie (2010)

Advanced Prediction Method in Efficient MPC Algorithm Based on Fuzzy Hammerstein Models

Piotr M. Marusak

Institute of Control and Computation Engineering, Warsaw University of Technology,
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland
P.Marusak@ia.pw.edu.pl

Abstract. An advanced prediction method utilizing fuzzy Hammerstein models is proposed in the paper. The prediction has such a form that the Model Predictive Control (MPC) algorithm using it is formulated as a numerically efficient quadratic optimization problem. The prediction is described by relatively simple analytical formulas. The key feature of the proposed prediction method is the usage of values of the future control changes which were derived by the MPC algorithm in the last iteration. Thanks to such an approach the MPC algorithm using the proposed method of prediction offers very good control performance. It is demonstrated in the example control system of a nonlinear control plant with significant time delay that the obtained responses are much better than those obtained in the standard MPC algorithm based on a linear process model.

Keywords: fuzzy control, fuzzy systems, predictive control, nonlinear control, constrained control.

1 Introduction

The control signals in the MPC algorithms are generated using prediction of the process behavior derived utilizing a model of the control plant [2,6,12,14]. Different models can be used to obtain the prediction. In the standard MPC algorithms linear process models are used. However, if the control plant is nonlinear such an approach can bring inefficient results. One of the class of nonlinear models are Hammerstein models [5]. In these models the nonlinear static part precedes the linear dynamic part. It is assumed that as the static part of the Hammerstein model the fuzzy Takagi–Sugeno (TS) model is used. It is because fuzzy TS models offer many advantages [11,13], like e.g. relative easiness of model identification, relatively small number of rules needed to describe even highly nonlinear functions and relatively simple obtaining of linear approximation using just the fuzzy reasoning. Moreover, it is assumed that the dynamic part of the Hammerstein model considered in the paper has the form of the step response.

Direct usage of a nonlinear process model in the MPC algorithm leads to its formulation as a nonlinear, nonquadratic, often nonconvex optimization problem, which must be solved in each iteration of the algorithm. Despite improved

versions of procedures solving the nonlinear optimization problems are designed (see e.g. [3] for modifications of particle swarm optimization and of genetic algorithms, taking into account properties of modern CPUs), nonlinear, nonconvex optimization has serious drawbacks. During solving an optimization problem of such kind time needed to find the solution is hard to predict. There is also problem of local minima. Moreover, in some cases numerical problems may occur. The drawbacks of the MPC algorithms formulated as nonlinear optimization problems caused that usually MPC algorithms utilizing a linear approximation of the control plant model, obtained at each iteration, are used [7,8,9,10,14]. Such algorithms are formulated as the standard quadratic programming problems.

The method of prediction proposed in the paper uses an approximation of the fuzzy Hammerstein model of the process. It is better method than the one proposed in [9] because the proposed prediction is obtained using not only the original fuzzy Hammerstein model and its linear approximation but it also uses values of the future control changes derived by the MPC algorithm in the previous iteration. Thanks to such an approach the obtained prediction is closer to the one obtained using the nonlinear process model only. At the same time as the prediction method exploits structure of the model it can be obtained relatively easy.

In the next section the general idea of the MPC algorithms is described. In Sect. 3 MPC algorithms based on linear models are described. Section 4 details the method of prediction generation utilizing the fuzzy Hammerstein model. The efficacy of the MPC algorithm which uses the proposed method of prediction is illustrated by the example results presented in Sect. 5. The last section summarizes the paper.

2 Model Predictive Control Algorithms – A General Idea

The MPC algorithms during its operation use prediction of the future process behavior many sampling instants ahead. The prediction is obtained utilizing a model of the control plant. Future values of the control signal are calculated in such a way that the prediction fulfills assumed criteria. These criteria are used to formulate an optimization problem which is solved at each iteration of the algorithm. The optimization problem has usually the following form [2,6,12,14]:

$$\min_{\Delta \mathbf{u}} \{ J_{\text{MPC}} = (\bar{\mathbf{y}} - \mathbf{y})^T \cdot (\bar{\mathbf{y}} - \mathbf{y}) + \Delta \mathbf{u}^T \cdot \Lambda \cdot \Delta \mathbf{u} \} \quad (1)$$

subject to:

$$\Delta \mathbf{u}_{\min} \leq \Delta \mathbf{u} \leq \Delta \mathbf{u}_{\max} \quad , \quad (2)$$

$$\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max} \quad , \quad (3)$$

$$\mathbf{y}_{\min} \leq \mathbf{y} \leq \mathbf{y}_{\max} \quad , \quad (4)$$

where $\bar{\mathbf{y}} = [\bar{y}_k, \dots, \bar{y}_k]^T$ is the vector of length p , \bar{y}_k is a set-point value, $\mathbf{y} = [y_{k+1|k}, \dots, y_{k+p|k}]^T$, $y_{k+i|k}$ is a value of the output for the $(k+i)^{\text{th}}$ sampling instant, predicted at the k^{th} sampling instant, $\Delta \mathbf{u} = [\Delta u_{k+1|k}, \dots, \Delta u_{k+s-1|k}]^T$,

$\Delta u_{k+i|k}$ are future changes in manipulated variable, $\mathbf{u} = [u_{k+1|k}, \dots, u_{k+s-1|k}]^T$, $\mathbf{A} = \lambda \cdot \mathbf{I}$ is the $s \times s$ matrix, $\lambda \geq 0$ is a tuning parameter, p and s denote prediction and control horizons, respectively; $\Delta \mathbf{u}_{\min}$, $\Delta \mathbf{u}_{\max}$, \mathbf{u}_{\min} , \mathbf{u}_{\max} , \mathbf{y}_{\min} , \mathbf{y}_{\max} are vectors of lower and upper limits of changes and values of the control signal and of the values of the output variable, respectively. The vector of optimal changes of the control signal is the solution of the optimization problem (14). From this vector, the first element, i.e. $\Delta u_{k|k}$ is applied in the control system and the algorithm goes to the next iteration.

The predicted values of the output variable $y_{k+j|k}$ are derived using the dynamic control plant model. If this model is nonlinear then the optimization problem (14) is, in general, nonconvex, nonquadratic, nonlinear, hard to solve optimization problem. Examples of such algorithms are described e.g. in [14].

3 MPC Algorithms Based on Linear Models

If in the MPC algorithm a linear model is used then the superposition principle can be applied and the prediction \mathbf{y} is described by the following formula [6][12][14]:

$$\mathbf{y} = \tilde{\mathbf{y}} + \mathbf{A} \cdot \Delta \mathbf{u} \quad (5)$$

where $\tilde{\mathbf{y}} = [\tilde{y}_{k+1|k}, \dots, \tilde{y}_{k+p|k}]^T$ is a free response of the plant which contains future values of the output variable calculated assuming that the control signal does not change in the prediction horizon; $\mathbf{A} \cdot \Delta \mathbf{u}$ is the forced response which depends only on future changes of the control signal (decision variables);

$$\mathbf{A} = \begin{bmatrix} a_1 & 0 & \dots & 0 & 0 \\ a_2 & a_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_p & a_{p-1} & \dots & a_{p-s+2} & a_{p-s+1} \end{bmatrix} \quad (6)$$

is a matrix, called the dynamic matrix, composed of coefficients of the control plant step response a_i .

After application of the prediction (5) to the performance function from the optimization problem (1) one obtains:

$$J_{\text{LMPC}} = (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A} \cdot \Delta \mathbf{u})^T \cdot (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A} \cdot \Delta \mathbf{u}) + \Delta \mathbf{u}^T \cdot \mathbf{A} \cdot \Delta \mathbf{u} \quad (7)$$

Note that the performance function (7) depends quadratically on decision variables $\Delta \mathbf{u}$. Moreover, the prediction (5) is applied in the constraints (4). Thus, all constraints depend linearly on decision variables. As a result, the optimization problem (14) becomes a standard linear–quadratic programming problem.

4 MPC Algorithm Based on Fuzzy Hammerstein Models

4.1 Model of the Process

It is assumed that in the Hammerstein process model (in which a nonlinear static part is followed by a linear dynamic part) the static part has the form of a fuzzy Takagi–Sugeno model:

$$z_k = f(u_k) = \sum_{j=1}^l w_j(u_k) \cdot z_k^j = \sum_{j=1}^l w_j(u_k) \cdot (b_j \cdot u_k + c_j) , \quad (8)$$

where z_k is the output of the static block, $w_j(u_k)$ are weights obtained using fuzzy reasoning, z_k^j are outputs of local models in the fuzzy static model, l is the number of fuzzy rules in the model, b_j and c_j are parameters of the local models. Moreover, it is assumed that the dynamic part of the model has the form of the step response:

$$\hat{y}_k = \sum_{n=1}^{p_d-1} a_n \cdot \Delta z_{k-n} + a_{p_d} \cdot z_{k-p_d} , \quad (9)$$

where \hat{y}_k is the output of the fuzzy Hammerstein model, a_i are coefficients of the step response, p_d is the horizon of the process dynamics (equal to the number of sampling instants after which the step response can be assumed settled).

4.2 Generation of the Free Response

The very basic idea of the approach proposed in [9] is to use the nonlinear model (9) to obtain the free response for the whole prediction horizon (assuming that the control signal will not change). The approach proposed in the current paper consists in utilization of future control increments derived by the MPC algorithm in the last sampling instant during calculation of the free response. Thus, it is assumed that future control values can be decomposed into two parts:

$$u_{k+i|k} = \check{u}_{k+i|k} + u_{k+i|k-1} , \quad (10)$$

where $\check{u}_{k+i|k}$ can be interpreted as the correction of the control signal $u_{k+i|k-1}$ obtained in the last $(k-1)^{\text{st}}$ iteration of the MPC algorithm. Analogously, the future increments of the control signal will have the following form:

$$\Delta u_{k+i|k} = \Delta \check{u}_{k+i|k} + \Delta u_{k+i|k-1} . \quad (11)$$

The output of the model (9) in the i^{th} sampling instant is described by the following formula:

$$\hat{y}_{k+i} = \sum_{n=1}^i a_n \cdot \Delta z_{k-n+i|k} + \sum_{n=i+1}^{p_d-1} a_n \cdot \Delta z_{k-n+i} + a_{p_d} \cdot z_{k-p_d+i} . \quad (12)$$

In (12) the first component depends on future action whereas the next ones depend on past control actions. Taking into account the decomposition of the input signal (10), (12) can be rewritten as:

$$\hat{y}_{k+i} = \sum_{n=1}^i a_n \cdot \Delta \check{z}_{k-n+i|k} + \sum_{n=1}^i a_n \cdot \Delta z_{k-n+i|k-1} + \sum_{n=i+1}^{p_d-1} a_n \cdot \Delta z_{k-n+i} + a_{p_d} \cdot z_{k-p_d+i} , \quad (13)$$

where $\Delta z_{k+i|k-1} = z_{k+i|k-1} - z_{k+i-1|k-1}$; $z_{k+i|k-1} = f(u_{k+i|k-1})$ and $\tilde{z}_{k+i|k-1} = z_{k+i|k} - z_{k+i|k-1}$. In (13) the second component is known and can be included in the free response of the control plant. Therefore, after taking into consideration the estimated disturbances (assumed the same for all instants in the prediction horizon — a DMC-type model of disturbances):

$$d_k = y_k - \hat{y}_k, \quad (14)$$

the final formula describing the elements of the free response will have the following form:

$$\tilde{y}_{k+i|k} = \sum_{n=1}^i a_n \cdot \Delta z_{k-n+i|k-1} + \sum_{n=i+1}^{p_d-1} a_n \cdot \Delta z_{k-n+i} + a_{p_d} \cdot z_{k-p_d+i} + d_k. \quad (15)$$

Thus, thanks to the form of the utilized fuzzy Hammerstein model the analytical equations describing the free response were obtained.

4.3 Generation of the Dynamic Matrix

Next, at each iteration of the algorithm the dynamic matrix can be easily derived using a linear approximation of the fuzzy Hammerstein model (9) [9]:

$$\hat{y}_k = dz_k \cdot \left(\sum_{n=1}^{p_d-1} a_n \cdot \Delta u_{k-n} + a_{p_d} \cdot u_{k-p_d} \right), \quad (16)$$

where dz_k is a slope of the static characteristic near the z_k . It can be calculated numerically using the formula

$$dz_k = \frac{\sum_{j=1}^l (w_j(u_k + du) \cdot (b_j \cdot (u_k + du) + c_j) - w_j(u_k) \cdot (b_j \cdot u_k + c_j))}{du}, \quad (17)$$

where du is a small number. The dynamic matrix will be thus described by the following formula:

$$\mathbf{A}_k = dz_k \cdot \begin{bmatrix} a_1 & 0 & \dots & 0 & 0 \\ a_2 & a_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_p & a_{p-1} & \dots & a_{p-s+2} & a_{p-s+1} \end{bmatrix}. \quad (18)$$

4.4 Formulation of the Optimization Problem

Finally, the prediction can be obtained using the free response (15) and the dynamic matrix (18):

$$\mathbf{y} = \tilde{\mathbf{y}} + \mathbf{A}_k \cdot \Delta \tilde{\mathbf{u}}, \quad (19)$$

where $\Delta\tilde{\mathbf{u}} = [\Delta\tilde{u}_{k+1|k}, \dots, \Delta\tilde{u}_{k+s-1|k}]^T$. After application of prediction (19) to the performance function from (1) one obtains:

$$J_{\text{FMPC}} = (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A}_k \cdot \Delta\tilde{\mathbf{u}})^T \cdot (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A}_k \cdot \Delta\tilde{\mathbf{u}}) + \Delta\mathbf{u}^T \cdot \mathbf{A} \cdot \Delta\mathbf{u} . \quad (20)$$

where $\Delta\mathbf{u} = \Delta\tilde{\mathbf{u}} + \Delta\mathbf{u}^{\mathcal{P}}$, $\Delta\mathbf{u}^{\mathcal{P}} = [\Delta u_{k|k-1}, \dots, \Delta u_{k+s-2|k-1}, 0]^T$; compare with (11). The prediction (19) is also used in constraints (4), formulas (10) and (11) are used to modify the constraints (2) and (3) respectively. Then, the linear–quadratic optimization problem with the performance function (20), constraints (2)–(4) and the decision variables $\Delta\tilde{\mathbf{u}}$ is solved at each iteration in order to derive the control signal.

Remark. It is also possible to use slightly modified performance function in which in the second component, only corrections of the control changes $\Delta\tilde{\mathbf{u}}$ are panelized, i.e.

$$J_{\text{FMPCv2}} = (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A}_k \cdot \Delta\tilde{\mathbf{u}})^T \cdot (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A}_k \cdot \Delta\tilde{\mathbf{u}}) + \Delta\tilde{\mathbf{u}}^T \cdot \mathbf{A} \cdot \Delta\tilde{\mathbf{u}} . \quad (21)$$

Such a modification, however, causes that the meaning of the tuning parameter λ is different than in the classical performance function. As a consequence, the algorithm with the modified performance function generates faster responses. It will be demonstrated in the next section.

5 Simulation Experiments

5.1 Control Plant

The control plant under consideration is an ethylene distillation column DA–303 from petrochemical plant in Plock. It is a highly nonlinear plant with a large time delay. In the Hammerstein model of the plant (Fig. 1) the static part is modeled by means of the Takagi–Sugeno model with three local models ($l = 3$) of the form:

$$z_k^j = b_j \cdot u_k + c_j , \quad (22)$$

where $b_1 = -2222.4$, $b_2 = -1083.2$, $b_3 = -534.4$, $c_1 = 9486$, $c_2 = 4709.3$, $c_3 = 2408.7$; values of these parameters as well as the assumed membership functions, shown in Fig. 2, were obtained using a heuristic approach based on analysis of the steady–state characteristic of the control plant. The output of the plant y is the impurity of the product. The manipulated variable u is the reflux. The higher the reflux is the purer product is obtained. During experiments it was assumed that the reflux is constrained $4.05 \leq u \leq 4.4$.

5.2 Results

Three MPC algorithms were designed:

- the NMPC one (with nonlinear optimization),
- the LMPC one (based on a linear model) and

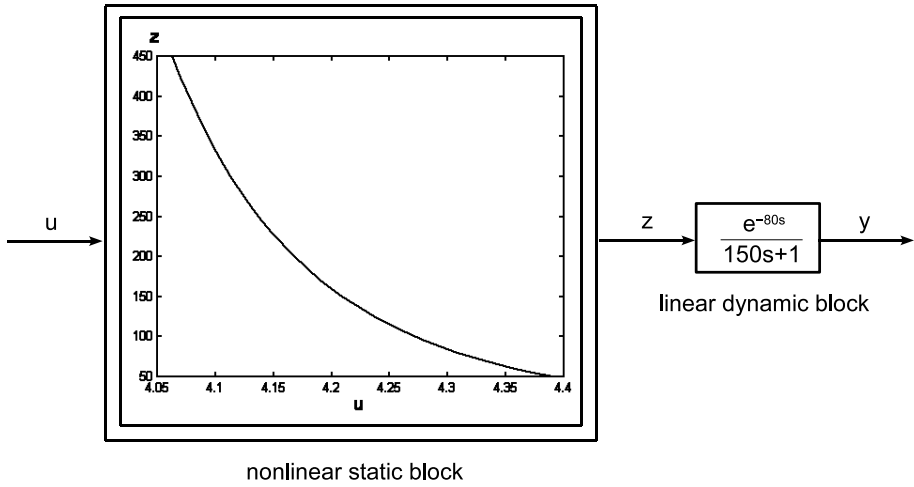


Fig. 1. Hammerstein model of the control plant

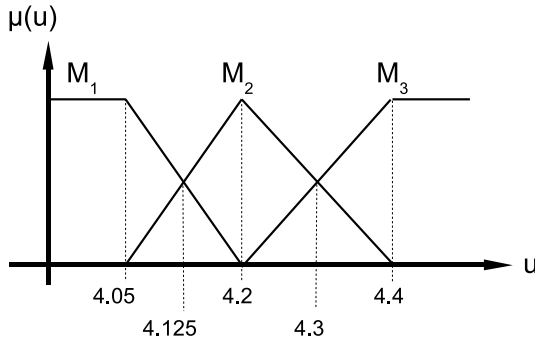


Fig. 2. Membership functions of the static part of the Hammerstein model

- the FMPC one (using the proposed method of prediction generation, based on the fuzzy Hammerstein model). Both versions of the algorithm were tested: the first one (FMPCv1) with the classical performance function (20) and the second one (FMPCv2) with the modified performance function (21).

The sampling period was assumed equal to $T_s = 20$ min, the prediction horizon $p = 44$, the control horizon $s = 20$ and the weighting coefficient $\lambda = 10^7$.

Example responses, obtained for $\lambda = 10^7$ are shown in Fig. 3. The response obtained with the LMPC algorithm to the set-point change to $\bar{y}_3 = 400$ ppm is unacceptable. The control system is very close to the boundary of stability. Moreover, the overshoot in response to the set-point change to $\bar{y}_2 = 300$ ppm is very big. On the contrary, both versions of the FMPC algorithm work well

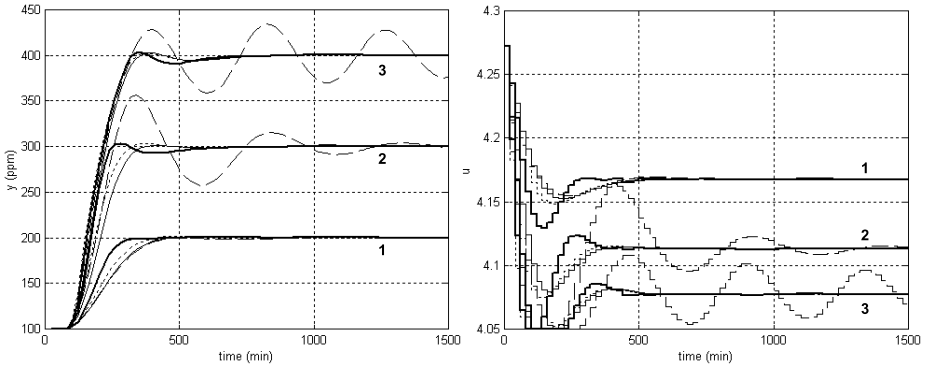


Fig. 3. Responses of the control systems to the change of the set–point values to $\bar{y}_1 = 200$ ppm, $\bar{y}_2 = 300$ ppm and $\bar{y}_3 = 400$ ppm, $\lambda = 10^7$; NMPC – dotted lines, FMPCv1 – thin solid lines, FMPCv2 – thick solid lines, LMPC – dashed lines; left – output signal, right – control signal

for all the set–point values; all responses have the same shape thanks to skillful application of the fuzzy (nonlinear) model.

The responses obtained with the FMPCv1 algorithm are very close to those obtained with the NMPC algorithm. It is, however, good to note that when the λ parameter was decreased (not too much – just to $\lambda = 10^6$), numerical problems occurred in the NMPC algorithm. These problems were not observed in the FMPC algorithm in which the control signal is obtained after solving the numerically robust linear–quadratic optimization problem. The fastest responses

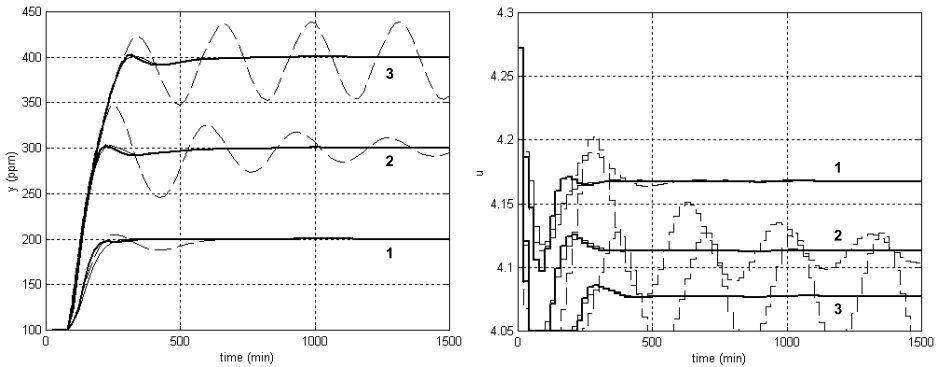


Fig. 4. Responses of the control systems to the change of the set–point values to $\bar{y}_1 = 200$ ppm, $\bar{y}_2 = 300$ ppm and $\bar{y}_3 = 400$ ppm, $\lambda = 10^6$; FMPCv1 – thin solid lines, FMPCv2 – thick solid lines, LMPC – dashed lines; left – output signal, right – control signal

were obtained with the FMPCv2 algorithm (thick solid lines in Fig. 3). They are better even than those obtained with the NMPC algorithm. It should be however stressed that it is a result of assuming, in the optimization problem solved by the FMPCv2 algorithm, a slightly different performance function than in other tested algorithms.

After decrease of the tuning parameter value to $\lambda = 10^6$, both versions of the FMPC algorithm work faster (Fig. 4) than in the case when $\lambda = 10^7$ (Fig. 3). As in the previous experiment, the FMPCv2 algorithm works faster than the FMPCv1 one, when the set-point changes to $\bar{y}_1 = 200$ ppm. In the case of responses to set-point changes to $\bar{y}_2 = 300$ ppm and to $\bar{y}_3 = 400$ ppm the differences in operation of both versions of the FMPC algorithm are small. It is because the control signal, at the beginning of these responses is on the constraint. It should be once more stressed that for $\lambda = 10^6$, numerical problems occurred in the NMPC algorithm and therefore there are no corresponding responses shown in Fig. 4.

6 Summary

The method of advanced nonlinear prediction generation was proposed in the paper. It is based on the fuzzy Hammerstein model of the process. The nonlinear model and values of the future changes of the control signal, calculated in the last iteration by the MPC algorithm, are used to derive the free response of the control plant. The linear approximation of the model, which is easy to obtain thanks to the structure of the utilized fuzzy Hammerstein model, is used to calculate the influence of only corrections of the future control signal used to obtain the free response. As a result, the FMPC algorithm, based on the proposed method of prediction, is formulated as the linear-quadratic optimization problem and offers excellent control performance, outperforming the standard LMPC algorithms based on linear process models. Two versions of the FMPC algorithm give more freedom to a control system designer who can choose the best version for a given problem.

Acknowledgment. This work was supported by the Polish national budget funds for science 2009–2011.

References

1. Babuska, R., te Braake, H.A.B., van Can, H.J.L., Krijgsman, A.J., Verbruggen, H.B.: Comparison of intelligent control schemes for real-time pressure control. *Control Engineering Practice* 4, 1585–1592 (1996)
2. Camacho, E.F., Bordons, C.: *Model Predictive Control*. Springer, Heidelberg (1999)
3. Chang, F.C., Huang, H.C.: A refactoring method for cache-efficient swarm intelligence algorithms. *Information Sciences* (2010) (in press), doi:10.1016/j.ins.2010.02.025
4. Fink, A., Fischer, M., Nelles, O., Isermann, R.: Supervision of nonlinear adaptive controllers based on fuzzy models. *Control Engineering Practice* 8, 1093–1105 (2000)

5. Janczak, A.: Identification of nonlinear systems using neural networks and polynomial models: a block-oriented approach. Springer, Heidelberg (2005)
6. Maciejowski, J.M.: Predictive control with constraints. Prentice Hall, Harlow (2002)
7. Marusak, P.: Advantages of an easy to design fuzzy predictive algorithm in control systems of nonlinear chemical reactors. *Applied Soft Computing* 9, 1111–1125 (2009)
8. Marusak, P.: Efficient model predictive control algorithm with fuzzy approximations of nonlinear models. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) ICANNGA 2009. LNCS, vol. 5495, pp. 448–457. Springer, Heidelberg (2009)
9. Marusak, P.: On prediction generation in efficient MPC algorithms based on fuzzy Hammerstein models. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010. LNCS (LNAI), vol. 6113, pp. 136–143. Springer, Heidelberg (2010)
10. Morari, M., Lee, J.H.: Model predictive control: past, present and future. *Computers and Chemical Engineering* 23, 667–682 (1999)
11. Piegat, A.: Fuzzy Modeling and Control. Physica-Verlag, Berlin (2001)
12. Rossiter, J.A.: Model-Based Predictive Control. CRC Press, Boca Raton (2003)
13. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. *IEEE Trans. Systems, Man and Cybernetics* 15, 116–132 (1985)
14. Tatjewski, P.: Advanced Control of Industrial Processes; Structures and Algorithms. Springer, London (2007)

Evolutionary Tuning of Compound Image Analysis Systems for Effective License Plate Recognition

Krzysztof Krawiec and Mateusz Nawrocki

Institute of Computing Science, Poznan University of Technology,
Piotrowo 2, 60965 Poznań, Poland

Abstract. This paper describes an evolutionary algorithm applied to tuning of parameters of license plate detection systems. We consider both simple and compound detection systems, where the latter ones consist of multiple simple systems fused by some aggregation operation (weighted sum or ordered weighted average). With the structure of a system given by a human and fixed, we perform an evolutionary search in the space of possible parameter combinations. Several simple and compound structures are considered and verified experimentally on frame collections taken from highly heterogeneous video sequences acquired in varying conditions. The obtained results demonstrate that all considered systems can be effectively tuned using evolutionary algorithm, and that compound systems can outperform the simple ones.

Keywords: Evolutionary computation, evolutionary image analysis, pattern recognition, mixtures of experts, license plate recognition.

1 Detection and Recognition of License Plates

License plate recognition (plate recognition for short) is a common benchmark for pattern recognition and computer vision systems, and one of their most frequent real-world applications. Typical use cases of such systems include authorization of entry for parking lots and gated blocks-of-flats, where the vehicle is close to the camera, the car is not moving (or close to still), and (sometimes) the lighting is partially controlled.

In this study, we focus on a more demanding scenario, where the working conditions of the system are much more unconstrained. This setup is more characteristic to CCTV monitoring in urban areas. Most importantly, the camera used in the experimental phase observes the moving vehicles from a relatively long distance. As a consequence, the observed projected dimensions of the plates are much smaller and the images can be distorted by motion blur and perspective projection. Also, nothing is assumed about the lighting conditions and the quality and state of the plates themselves (the presence of dust, for example). We allow also for the presence of multiple vehicles in the field of view of the camera.

The major contribution of this paper is a compound license plate recognition system that relies on the mixture of experts design pattern and employs evolutionary algorithm to tune the parameters of its particular components. After reviewing selected past work on this topic in section 2, we outline the overall architecture of the system that serves as a framework for this study (Section 3). In Section 4 we describe the elementary components of the considered compound detection systems. Section 5 details on the setup and results of experimental evaluation of the considered recognition systems, in particular on the approach to evolutionary tuning of its parameters. In the final Section 7 we discuss the results and point out the possible further research directions.

2 Related Work

Due to numerous publications, a complete review of all past work done in the area of license plate recognition is beyond the scope of this paper. Former research on this topic engaged various paradigms from computational intelligence, including artificial neural networks, fuzzy logic, and evolutionary computation. For instance, in [14], a fuzzy logic approach has been applied to the problem of number plate recognition. In [7] the author presents the survey of many techniques used in automatic plate recognition systems. Techniques for every stage of recognition process are discussed there: edge detection, image projection, statistical analysis and deskewing mechanism for number plate area detection, horizontal projection for plate segmentation, artificial neural networks for character recognition, and syntactic analysis. The reader interested in these topics is recommended to refer to this review.

There are quite numerous accounts on the use of evolutionary computation for plate recognition. In [12] authors use immune and genetic algorithms to acquire the parameters for the initial step of plate recognition. Thresholds and weights of the neural network are optimized by genetic algorithm in [10]. In [5], genetic algorithm is used to determine the region that covers the license plate.

The plate recognition systems presented in [11,13] can serve as another examples of approaches that could be compared side-by-side to the method presented in this paper. This applies also to many commercial solutions. Unfortunately, in most cases the methodology used and the values of performance indicators are the producers' secret, which renders such comparison difficult.

3 Architecture of the Complete System

The overall data flow in the system is unidirectional (bottom-up) and can be divided into four separate stages: motion segmentation (MS), plate detection (PD), character segmentation (CS), and character recognition (CR). The former three stages have been designed based exclusively on domain-specific knowledge and human experience; the last stage involves a powerful support vector machine classifier. In its current form, the method processes each video frame independently (apart from limited use of the previous frame in the MS stage).

The **motion segmentation** (MS) stage is responsible for constraining the system's region of interest (ROI) to those parts of the input frame that potentially represent moving vehicles. As the localization algorithms became efficient, this stage is omitted in the configuration considered in this paper.

The **plate detection** (PD) stage employs sophisticated filtering to determine the potential locations of license plates, called *plate candidates* in following. All the candidates are passed to the CS phase. This is important, not only because of the potential presence of objects that resemble plates, but also because a single frame may actually contain more than one plate, if more than one moving car may turn up in the field of view of the camera.

For each plate candidate returned by the PD stage, the **character segmentation** (CS) stage makes an attempt to segment it and produce a sequence isolated small images representing subsequent characters. Before segmentation, each candidate plate is deskewed. First, the most salient line within the region is found. It is always the top or bottom border of the plate. The slope of that line is then used to rotate the whole plate candidate.

Having deskewed the plate candidate, the CS proceeds to actual character segmentation, which is mostly focused on analyzing the horizontal profile or, in other words, a vertical shadowgraph of the plate candidate. The candidates whose horizontal profile does not resemble that of a typical plate are rejected. If no candidate's profile passes this test, the algorithm assumes that there is no plate in the frame and processing ends at this stage. Thus, this stage may in general produce more than one sequence of character images.

The subsequent **character recognition** stage (CR) processes independently each segmented character image provided by the previous stage. The pixels of character image are fed into support vector classifier (SVM, [9,2]), previously trained on a large collection of human-segmented characters belonging to 36 classes (26 uppercase Latin alphabet letters plus 10 digits). For each of the 36 decision classes, the SVM classifier returns a continuous value that reflects the likelihood of the character belonging to the class. The class with the highest likelihood determines the final decision of the classifier. The recognitions made by SVM for the subsequent characters are concatenated into one character string and returned as the final outcome of the method.

4 Using Mixtures of Experts for License Plate Detection

The concept of mixture of experts is founded on quite intuitive hypothesis that aggregation of multiple different subsystems (experts, predictors, classifiers) can perform better than each subsystem separately. Canonic instances of this paradigm are bagging and boosting algorithms in machine learning [4]. For some variants of this scheme, it has been even proven formally that combining multiple weak yet non-correlated classifiers leads in limit to a perfectly performing compound classifier. This feature has been exploited in many practical applications of machine learning and pattern recognition algorithms [6].

In this paper, we extend our former single-expert approach [8] by blending the mixture of experts paradigm with evolutionary tuning of the overall compound

Table 1. Summary of simple operators (single filters) and compound operators (aggregators) used in the experiment (see text for detailed description)

Operator	Description	Output type	Number of tunable parameters
M	Mask (convolution)	continuous	$4 + 15 \times 15 = 229$
C	Color filter	continuous	4
V	Variance thresholding	binary	5
P	Profile detector	continuous	21
WS()	Weighted Sum	binary	number of arguments+1
OWA()	Ordered Weighted Averaging	binary	number of arguments+1

system. There are two qualitatively different aspects that need to be considered when building a compound image analysis system: its structure (which determines the data flow in the system) and parameters. In this study, the structure of the system is given by the designer and remains fixed, so that the learning task is constrained to parameter optimization.

We assume that there are two types of blocks that our compound recognition systems are built of: *filters* and *aggregators*. Each filter takes a single image (typically the original RGB video frame) as an input and produces a single, same-sized, one-channel image at its output. An aggregator, on the other hand, accepts two or more identically sized single-channel images at input, and produces a single, same-sized, one-channel image at its output. Table 1 summarizes the filters and aggregators used in this paper.

The aggregators considered in the following experiment work pixel-wise, i.e., the intensity of a particular pixel in the output image depends only on intensities of the corresponding pixels in the input images. The first aggregator is Weighted Sum (WS), where the output pixel is simply sum of input pixels multiplied by scalar coefficients. The important characteristics of this filter is that weights are associated one-to-one to particular input images in a fixed way.

In the second aggregator, called Ordered Weighted Averaging (OWA, [13]), there is no such association. For a particular pixel coordinates, OWA first sorts descendingly the intensities of that pixel in the input images. Next, the intensities are multiplied by a vector of weights. The resulting dot product becomes the output value. In this way, OWA can pay different attention to particular input images when considering different pixels.

Both aggregators can have an arbitrary number of inputs, and with each of the inputs a single weight is associated. Additionally, each aggregator compares the obtained aggregated value to a threshold, which is also tunable. Therefore, the total number of parameters that determine the working of an aggregator is equal to the number of its inputs plus one. This is also the number of genes in individual's encoding that are required to encode an aggregator.

We use four filters, three of which, denoted by M, C, and V in Table 1, are quite generic. The first of them, called *Mask* (M) in the following, is a simple convolution of the input image with a mask. Both mask dimensions (width w

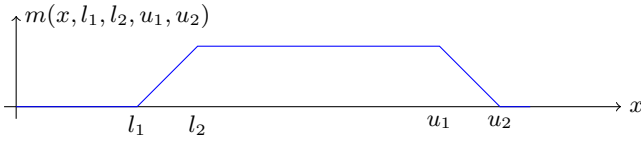


Fig. 1. Fuzzy membership function used in the Color filter

and height h) as well as its weights can be tuned by the search algorithm. When applied to an image, the mask is shifted horizontally and vertically with parameterized steps x_{inc} and y_{inc} . The upper limit on mask size is set to 15×15 . Therefore, the complete encoding of this filter in individual's genome requires $2 + 2 + 15 \times 15 = 229$ genes.

The *Color* filter (C) works in the HSV color space and imposes pixel-wise soft thresholding on color saturation and value (intensity):

$$g(x, y) = m(s(x, y), 0, 0, s_1, s_2) + m(v(x, y), v_1, v_2, 255, 255) \quad (1)$$

where g denotes the intensity of the output pixel, $s(x, y)$ and $v(x, y)$ are respectively the saturation and the value of pixel (x, y) in the input image, s_1 and s_2 are the (soft) thresholds for saturation, v_l and v_u are the thresholds for value, and $m(x, l_1, l_2, u_1, u_2)$ is a trapezoidal fuzzy membership function defined as in Fig. 1. To encode the parameters of this filter in individual's genome, one needs four genes, one for each of l_1, l_2, u_1 and u_2 .

The *Variance thresholding* filter (V) imposes crisp thresholding on the variance of intensity of the original image, calculated from $w \times h$ window, where $w \in [10, 100]$ and $h \in [5, 30]$ centered on the considered pixel. Similarly to the M filter, when applied to an image, the V filter is shifted horizontally and vertically with parameterized steps x_{inc} and y_{inc} . Thus, its encoding requires five genes (variables): 2 for w and h , 2 for x_{inc} and y_{inc} , and one for threshold.

The fourth filter, called *Profile* (P) in the following, stems from our former research on license plate recognition [8] and is more sophisticated and tailored specifically to the task of license plate detection. This filter's field of view is constrained to a single horizontal row of pixels of a certain length. The pixels are scanned from left to right, and 12 different statistical descriptors ($d_1 \dots d_{12}$), $d_i \in [0, 1]$ are collected from them. They take into account the characteristics of pixels' brightness distribution, like for example the difference between maximal and minimal brightness. Based on these statistics, the output of the filter is defined as $\prod_i d_i$. The detailed construction of this filter is beyond the scope of this paper. The encode of a particular instance of this filter in individual's genome requires 21 genes.

5 The Experiment

The primary objective of the experiment is to verify the usefulness of the mixture-of-experts paradigm for the task license plate detection by comparing selected compound recognition systems with the single filters.

The second aspect to investigate is the usefulness of evolutionary tuning of system parameters. We claim that evolutionary algorithm is an appropriate tool for that purpose, because the number of optimized parameters may be quite large here (particularly for the compound systems), and the performance of a system depends on its parameter setting in a complex, non-linear way. Also, the settings of particular parameters interact with each other (epistasis).

We use collection of 1233 frames of 160 different vehicles (mostly passenger cars) passing in the field of view of the camera, previously used in [8]. Each frame has been manually inspected and the actual (true) license number has been assigned to it. All frames have been acquired using the same stationary FireWire camera working with resolution 1280×960 pixels, located at 15-20 meters from the passing-by cars. The following discussion concerns the frames after the motion segmentation phase, which have typically VGA-comparable resolution. In these frames, the vehicles occupy on average 75% of the frame area, almost all of them in frontal view (only a few frames present rear view). The plates to be recognized have typically dimensions of 150×30 pixels, however, they are often far from being rectangular due to perspective projection and vehicle's tilt and yaw.

It should be emphasized that the prepared dataset has been acquired in realistic conditions and is highly heterogeneous: it comprises various lighting conditions (different time of the day, including backlight as well as plates directly exposed to sunlight), different weather conditions (both sunny and cloudy days), and with license plates subject to dirt and mounted at different heights relative to road level.

The experiment consisted in evolutionary tuning of parameters for different setups of simple and compound systems. We assume that no more than one instance of aggregator is used per setup, which implies 26 possible setups: 4 setups that use single filters and $2 \times (2^4 - 5) = 22$ non-trivial compound setups that involve single aggregator and two, three, or four different filters (an aggregator has to have at least two arguments for the setup to be non-trivial). If one allows using the same filter more than once in a setup, the number of compound setups increases to 26. For brevity, rather than considering all possible compound setups, in this paper we focused on selected compound setups only, summarized in Table 2. Note that different setups imply different numbers of tunable parameters, and therefore different dimensionality of the search space.

Let us also note that among the considered setups we include also one that involves a 'blocked' filter, by which we mean a filter evolved in a separate, earlier evolutionary run, after which its parameters have been fixed to *prevent* the further tuning by the next evolutionary process. This setup is in a sense incremental, as it attempts to build upon a previous, independent learning process.

Given the actual (true) plate location (rectangle) P_{act} and the n plate candidates P_i , $i = 1, \dots, n$, (also defined as rectangles), an individual's fitness is defined as the relative overlap of both rectangles:

$$\max_i \frac{area(P_{act} \cap P_i)}{area(P_{act} \cup P_i)} \quad (2)$$

Table 2. Summary of compound setups

Setup	Number of tunable parameters (genome length)
WS(P,P)	$21 + 21 + (2 + 1) = 45$
OWA(C,P-blocked)	$4 + 2 + 1 = 7$
OWA(C,P)	$4 + 21 + 2 + 1 = 28$
OWA(P,P)	$21 + 21 + 2 + 1 = 45$
OWA(M,C,V,P)	$229 + 4 + 5 + 21 + 4 + 1 = 264$

where \cap denotes intersection of rectangles. We average this indicator over the entire training set, composed of 100 images drawn randomly from our collection of frames (of which 97 contain any license plates). The same training set has been used in all evolutionary runs.

For each setup, we run generational evolution algorithm on population of 1000 individuals for 100 generations. Individuals are represented as vectors of variables (floating-point numbers) that correspond to parameters used by particular setups. For breeding, both parent solutions are selected independently from the previous population via tournament of size 5 and recombined using one-point crossover. In the resulting offspring, genes (variables) undergo mutation with probability 0.05 per gene. Mutation is Gaussian and multiplicative, affecting more the genes that have large values. If the mutated value violates the $[0, 1]$ bounds, it is clamped.

In Table 3 we report the fitness of the best-of-run individual for each setup, and the performance on a disjoint test set of 100 images (of which 98 contain plates). Though the performance of compound detectors is on average better, none of them clearly outperforms the P filter.

Figure 2 presents exemplary results of the plate detection process carried out by the best-of-run individual of the WS(P,C) setup for one of the frames. Light-colored regions indicate the locations where the filters' belief in plate presence is higher. A closer inspection of images produced by constituent filters reveals that their fusion is synergetic, i.e., they complement each other in a way which leads to better performance of the overall compound detection system.

Table 3. Fitness of the best-of-run individuals for simple (left) and compound (right) setups, for the training set and the testing set

Setup	Training set	Testing set	Setup	Training set	Testing set
M	0.5439	0.4128	WS(P,P)	0.6805	0.5512
C	0.6263	0.5335	OWA(C,P-blocked)	0.7595	0.6437
V	0.3300	0.3162	OWA(C,P)	0.6554	0.5249
P	0.7536	0.6416	OWA(P,P)	0.7402	0.6148
			OWA(M,C,V,P)	0.6276	0.5212



Fig. 2. The process of recognition implemented by the best-of-run individual of WS(P,C) setup: the output of the C filter (top left), the output of the P filter (top right), and the aggregated confidence image produced by the WS aggregator (bottom left), and the final detection outcome (bottom right) with the detected localizations

6 Driving Evolution by Recognition Accuracy

As the next step, we embedded the best evolved plate localization subsystems in the complete plate recognition system and tested its recognition rate. The performance turned out to be disappointing, with the number of erroneous plate readings far too high for practical use. This suggests that the evolutionary algorithm found a way to maximize the overlap between the actual plate location and the found plate candidates, but for some reasons this capability has not been reflected in the actual plate recognition rate.

This observation inclined us to redefine the learning task and guide the simulated evolution using the actual recognition rate. To this aim, we designed a new fitness function based directly on the detected character sequences:

$$\frac{1}{n} \sum_{i=1}^n \frac{d_{max} - \min(d_{max}, d(s, s_i))}{d_{max}} \tag{3}$$

where n is the number of plates in the training set, s is the character string representing the plate number as read by the recognition system, s_i is the actual plate number present in the i th training frame, and d is the Levenshtein distance metric $d_{max} = 5$, so the maximal distance that positively contributes to fitness is 4 (most plate numbers used here had 7 characters). If $d(s, s_i) \geq 5$, an individual scores 0 for the frame.

Table 4. Distribution of Levenshtein distance d

Levenshtein distance d	0	1	2	3	4	5	$d \geq 5$	Total
Training set images	60	25	5	0	0	4	3	97
Test set images	38	23	6	0	2	2	27	98
Test set images (corrected)	57	8	2	1	1	2	27	98



Fig. 3. Comparison of the 0 (zero, left) and O (right) characters

The experiment was conducted for three configurations that appeared to work best in the previous test: the simple P and C configurations, and the combined WS(C, P) configuration. Apart from the new definition of fitness function, we used the same parameter setting as in Section 5. Of the three considered configurations, WS(C,P) turned out to produce the best final training set fitness of 0.85, with over 61.9% of plates' images perfectly detected and recognized. This result confirms our working hypothesis that synergy between two or more localizers is possible.

Table 4 presents the distributions of Levenshtein distance for the training and test set. For the test set, the share of perfect recognitions drops to 38.8%. Clearly, most of the errors boil down to single-character mistakes, implying that the plate detection phase works well, as the remaining characters are correctly recognized. This suggests that, given better character recognizer, further improvements are likely. Thus, we analyzed the statistics of errors committed by the recognizer and found out that the majority of errors consist in confusing the 0 (zero) and O characters. This should not come as a surprise, given how similar these characters are in Polish license plates, as demonstrated in Fig. 3. These characters differ only in aspect ratio, which can be easily distorted by the projection of plate image onto camera sensor. Therefore, perfect discrimination of these decision classes is impossible without help of some additional information, like syntactic rules (e.g., 'license plate cannot start with a numeral'). If we accept this fact and treat the 0 and O characters exchangeably by merging them into one decision class, the structure of errors changes to the one shown in the bottom row of Table 4, meaning 58.2% of perfect recognitions. Though this result may still seem far from perfect, one has to take into account that the same vehicle is typically visible in a few consecutive frames, so the recognition rate can be potentially boosted by aggregating recognitions obtained for multiple frames. Given high performance of our approach (about 40 frames per second), this could be done at no extra cost.

7 Summary

In this study, an evolutionary algorithm proved useful for optimization of parameters of simple and compound licence plate recognition systems. The attained recognition rate turned out to be much higher than that obtained by manual tuning, and the compound recognition systems that aggregate the outputs of multiple heterogeneous or homogeneous subsystems improve the detection rate, compared to single filters.

Acknowledgments. Work supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°218086.

References

1. Abdullah, S., Khalid, M., Yusof, R., Omar, K.: License plate recognition using multi-cluster and multilayer neural networks 1, 1818–1823 (2006)
2. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
3. For, W.K., Leman, K., Eng, H.L., Chew, B.F., Wan, K.W.: A multi-camera collaboration framework for real-time vehicle detection and license plate recognition on highways, pp. 192–97 (June 2008)
4. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P.M.B. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995)
5. Ji-yin, Z., Rui-rui, Z., Min, L., Yin, L.: License plate recognition based on genetic algorithm. In: 2008 International Conference on Computer Science and Software Engineering, vol. 1, pp. 965–968 (2008)
6. Krawiec, K., Kukawka, B., Maciejewski, T.: Evolving cascades of voting feature detectors for vehicle detection in satellite imagery. In: IEEE Congress on Evolutionary Computation, July 18–23, pp. 2392–2399. IEEE Press, Barcelona (2010)
7. Martinsky, O.: Algorithmic and mathematical principles of anpr systems (2007)
8. Nawrocki, M., Krawiec, K.: A robuts method for real-time detection and recognition of licence pla tes. In: IEEE International Conference on Multimedia Communications, Services, and Security (MCSS 2010), pp. 171–175 (2010)
9. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods – Support Vector Learning. MIT Press, Cambridge (1998)
10. Sun, G., Zhang, C., Zou, W., Yu, G.: A new recognition method of vehicle license plate based on genetic neural network. In: 2010 the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 1662–1666 (2010)
11. Tseng, P.-C., Shiung, J.-K., Huang, C.-T., Guo, S.-M., Hwang, W.-S.: Adaptive car plate recognition in qos-aware security network. In: SSIRI 2008: Proceedings of the 2008 Second International Conference on Secure System Integration and Reliability Improvement, pp. 120–127. IEEE Computer Society, Washington, DC, USA (2008)
12. Wang, F., Zhang, D., Man, L.: Comparison of immune and genetic algorithms for parameter optimization of plate color recognition. In: 2010 IEEE International Conference on Progress in Informatics and Computing (PIC), vol. 1, pp. 94–98 (2010)
13. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Transactions on Systems, Man and Cybernetics 18(1) (1988)
14. Zimic, N., Ficzko, J., Mraz, M., Virant, J.: The fuzzy logic approach to the car number plate locating problem. In: IASTED International Conference on Intelligent Information Systems, p. 227 (1997)

Investigation of Self-adapting Genetic Algorithms Using Some Multimodal Benchmark Functions

Magdalena Smętek and Bogdan Trawiński

Wrocław University of Technology, Institute of Informatics,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{magdalena.smetek,bogdan.trawinski}@pwr.wroc.pl

Abstract. Self-adaptive mutation, crossover, and selection were implemented and applied in three genetic algorithms. So developed self-adapting algorithms were then compared, with respect to convergence, with a standard genetic one, which contained constant rates of mutation and crossover. The experiments were conducted using five multimodal benchmark functions. The analysis of the results obtained was supported by nonparametric Friedman and Wilcoxon signed-rank tests. The algorithm employing self-adaptive selection revealed the best performance.

Keywords: self-adapting GA, self-adaptive selection, benchmark functions.

1 Introduction

Choosing right parameter values of variation operators (mutation and recombination), selection mechanism of parents and initial population is crucial when applying evolutionary algorithms (EA). It determines whether the algorithm will find a near-optimum solution and whether it will find such a solution efficiently [7].

A few taxonomies of parameter setting forms in EA have been introduced in [1], [7], [17]. Angeline [1] categorized adaptation by the level at which the adaptive parameters operate. He proposed three different adaptation level: population-level where parameters that are global to the population are adjusted, individual-level where changes affect each member of the population separately, and component-level where each component of each member may be modified individually.

Smith and Fogarty [17] distinguished three criteria which were the basis of division. The criteria are: what is being adapted, the scope of the adaptation, and the basis for change. The latter is further split into two categories: evidence upon which the change is carried out and the rule or algorithm that executes the change.

General taxonomy of parameter value setting was created by Eiben, Hinterding, and Michalewicz [7]. It distinguishes two major forms: parameter tuning and parameter control. The taxonomy was shown in Fig. 1. A fundamental difference between this two forms of parameter value setting is time of change parameter's values. In first case, determining proper values for the parameters takes place before running GA/EA. These values are not changed during the run, what contradicts the dynamic nature of GA/EA. In the second case parameter's values are adjusted during the execution. The authors specified three classes of parameters control: deterministic,

adaptive and self-adaptive. Deterministic techniques modify parameters according to some deterministic rules but without using any feedback from the optimization process. Adaptive methods use the feedback to adjust parameters, whereas by self-adaptive parameter control the parameters are encoded directly into the chromosome. Then, the parameters are subject to mutation and recombination process.

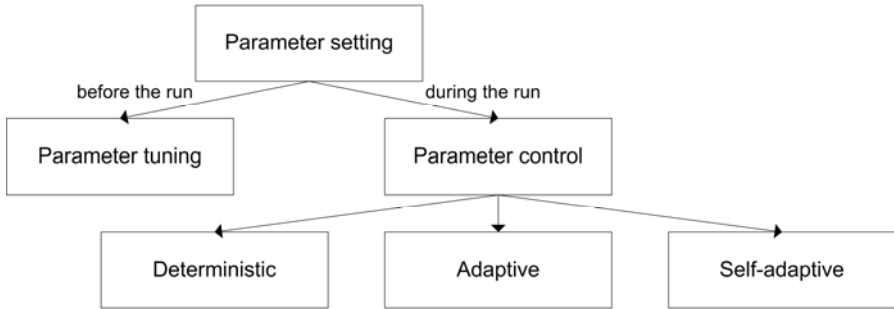


Fig. 1. General taxonomy of parameter in evolutionary computation [7]

Many parameter control method have been proposed [2], [3], [4], [5], [9], [12], [13], [15],[17]. Bäck [2] showed that at least a combination of one adaptive mutation rate per individual and extinctive selection is better than GA. Maruo et al. [12] also emphasized the benefits of using self-adaptation. They proposed method based on encoding in the chromosome mutation and crossover rates, crossover type, mutation step, and the size of tournament size. The chromosome takes part in mutation and crossover if rates encoded in chromosome are greater than real values bound with them in special matrixes. Schaffer and Morishima [15] presented punctuated crossover, which adapts the positions where crossover occurs, and they reported that it performed better than one-point crossover.

In many experiments benchmark functions are exploited to validate effectiveness and convergence of novel techniques and to compare them with other methods [6], [14], [18], [19], [20]. Yao [20] categorized them into three groups: unimodal functions, multimodal with many local minima, and multimodal with a few local minima. Multimodal functions especially with many local minima are often regarded as being difficult to find their optima.

Our former investigations on the use of evolutionary algorithms to learn the rule base and learn membership functions of fuzzy systems devoted to aid in real estate appraisal showed it is a laborious and time consuming process [10], [11]. Therefore, we intend to examine the usefulness of incorporating self-adapting techniques into our genetic fuzzy systems aimed to generate models for property valuation [8].

Studies presented here are a continuation of the our research presented in [16]. Another self-adapting methods of mutation and crossover were implemented and examined here. The goal of the study was to test the proposed methods using some selected multimodal benchmark functions.

2 Self-adapting Algorithms

A self-adaptive method was implemented employing a binary encoding of the chromosome, which, besides the solution, i.e. an argument or arguments of a given benchmark function, comprises mutation and crossover rates, number of crossover points and number of genes to mutation thereby making them subject to evolution. The solution is represented with the accuracy of six decimal places, whereas the rates of both mutation and crossover and number of two decimal places. The mutation and crossover rates can take real values from the ranges of [0.0,0.3] and [0.0,1.0] respectively. Number of crossover points can take integer values from the range [1,7], and number of genes to mutation can take integer value from ranges [0-15] percent of chromosome's length. To encode a real value X in the binary chromosome Y formula (1) was used:

$$Y = [X \cdot 10^d]_2 \quad (1)$$

where d denotes an accuracy and $[Q]_2$ means the conversion of Q to the binary system. In turn, to obtain a real value from the chromosome formula (2) was applied:

$$X = [Y]_{10}/10^d \quad (2)$$

where $[Q]_{10}$ denotes the conversion of Q to the decimal system. According to the above rules 5 genes are required for the mutation rate and 7 genes for the crossover rate. To encode integer values for number of crossover points 3 genes are required and 4 genes are required to encode number of genes to mutate.

Three self-adapting algorithms were used in experiment: SAAVG, SAMCP and SAS.

1) *SAAVG. Genetic Algorithm with Averaged Self-adapting Mutation and Crossover.* The algorithm adapts mutation rate, crossover rate and number of crossover points, so that 15 extra genes were added to each chromosome in the population: five per mutation rate, seven per crossover rate and three to encode the number of crossover points. Fig. 2 demonstrates the chromosome.

Solution	Mutation rate	Crossover rate	No. of crossover points
----------	---------------	----------------	-------------------------

Fig. 2. Chromosome with self-adaptive parameters for SAAVG

The algorithm works similarly to GA, but after selection process the mutation rate and crossover rate are calculated as an average of mutation and crossover rates which were extracted from each chromosome in population and decoded to real value. Crossover differs because of variable number of crossover points. Fitness function (FF) is counted for each pair of chromosomes which takes part in the crossover. Then standard crossover placed with the number of breaking points which was extracted from the chromosome with lower value of FF. The schema of self-adaptive crossover for SAAVG was depicted in Fig. 3.

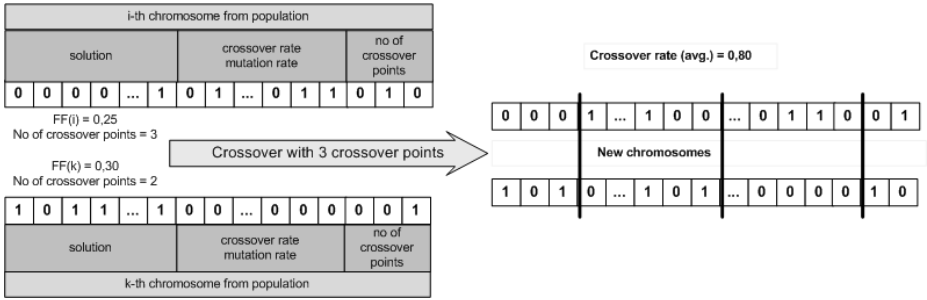


Fig. 3. Self-adaptive crossover for SAAVG

2) *SAMCP. Genetic Algorithm with Self-adapting Mutation and Crossover* with encoded number of crossover points and genes to be mutated. 19 extra genes were added to each chromosome to encode mutation rate (5 genes), crossover rate (7 genes), number of crossover points (3 genes) and number of genes (in one chromosome) which undergo mutation (4 genes). Fig. 4 illustrates the chromosome.

Solution	Mutation rate	Crossover rate	No. of crossover points	No. of genes to mutation
----------	---------------	----------------	-------------------------	--------------------------

Fig. 4. Chromosome with self-adaptive parameters for SAMCP

The algorithm employs modified mutation and crossover in comparison with the classic GA. The self-adaptive mutation is illustrated in Fig. 5.

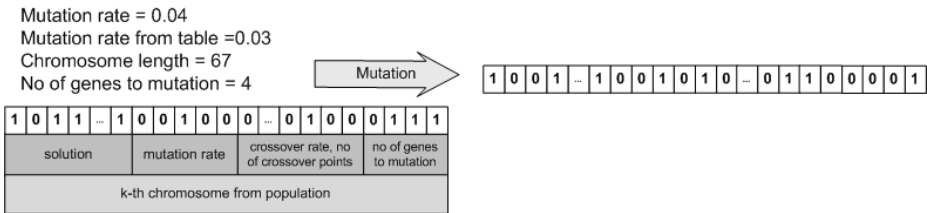


Fig. 5. Self-adapting mutation for SAMCP

Each chromosome from the population can be subject to mutation. A $N \times 1$ table with real, randomly selected values from the range of $[0.0,0.3]$ is created, where N is the number of chromosomes in the population. Each chromosome is connected with one real value in the table. The self-adaptation of the mutation proceeds as follows. For each chromosome from the population:

- Extract the genes representing the mutation rate from the chromosome.
- Calculate the value of mutation rate extracted from chromosome.
- If the value from the table is lower than the value of the mutation rate taken from the chromosome, then the chromosome participates in mutation and

number of genes to mutate is extracted from the chromosome. The genes are randomly selected and undergo mutation in a traditional way.

- The table remains unchanged during the run.

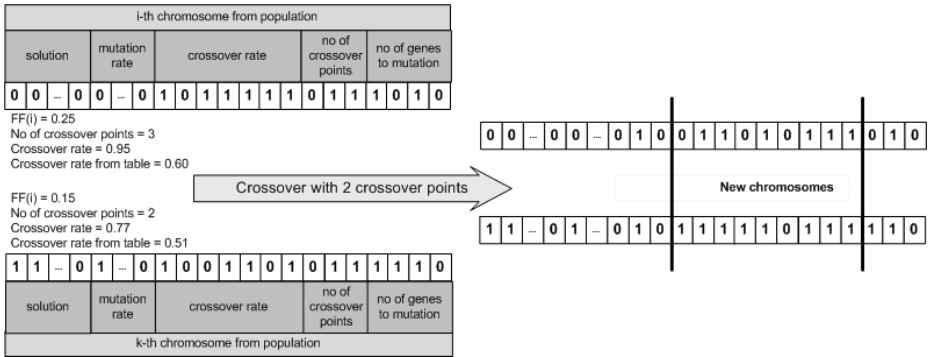


Fig. 6. Self-adaptive crossover for SAMCP

The self-adaptive crossover in SAMCP depicted in Fig. 6 is also different from a traditional GA crossover. A $N \times 1$ table with real, randomly selected values from the range of $[0.5, 1.0]$ is created, where N is the number of chromosomes in the population. Each chromosome is connected with one real value in the table. The self-adaptation of the crossover proceeds in the following way. For each chromosome from population:

- Extract the genes representing the crossover rate from the chromosome.
- Calculate the value of crossover rate extracted from the chromosome.
- If the value from the table is lower than the value of crossover rate from the chromosome, then the chromosome is selected to a classic crossover process.
- The table remains unchanged during the run,
- Chromosomes selected to crossover are randomly associated in pairs. Number of crossover points is equal to the number of crossover points extracted from the chromosome, which had lower fitness function value than the other chromosome in the pair.

3) SAS. Genetic Algorithm with Self-adapting Selection. This algorithm has been introduced in [16]. No extra genes are added to chromosome, only a solution is encoded in it, so that the chromosomes looked exactly like chromosomes in a classic GA. Self-adapting selection consists in the control of a population size. Each chromosome is connected with one integer value, which represents the aging level of the chromosome. At the beginning this value was set 3 for each chromosome. The self-adaptation of the selection proceeds as follows. For each chromosome from the population:

- Subtract 1 from the aging level.
- Add 1 to the aging level if the value of the chromosome fitness function is lower than median of the values of all chromosomes or subtract 1 from the aging level in the opposite case.
- Subtract 2 from the aging level if the population size has been increased 10 times and the fitness function of the chromosome is not in top 1000 values of the fitness function in the whole population.
- Remove from the population all chromosomes with the aging level lower or equal to zero.

This algorithm changes also mutation process because two parameters instead of mutation rate were used: number of chromosomes which undergo mutation and number of genes which undergo mutation in selected chromosomes. The first one was set to 0.15, what meant that 15% chromosomes took part in mutation. The second parameter was set to 1/3, what means that one third of genes in the chromosome undergoes mutation.

3 Plan of Experiments

The main goal of our experiment was to compare, in respect of a convergence, a classic GA with three self-adapting GAs. In the classic GA population size was set to 100, mutation rate was equal to 0.15, crossover rate to 0.80, and elite to one. Moreover selection with seven-chromosome tournament and crossover with two randomly determined breakpoints were applied.

Table 1. Benchmark functions used in experiments

Function	n	Domain	f_{min}
$f_1(x) = \sum_{i=1}^n [-x_i \sin(\sqrt{ x_i })]$	30	[-500,500]	-418.9829n
$f_2(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \frac{x_i}{\sqrt{i}} + 1$	30	[-600,600]	0
$f_3(x) = -20e^{-0.2\sqrt{\frac{\sum_{i=1}^n x_i^2}{n}}} - e^{\sum_{i=1}^n \cos(2\pi x_i)} + 20 + e$	30	[-1,-1]	0
$f_4(x_1, x_2) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	[-5,10] [015]	0.397887
$f_5(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$	2	[-6,6]	0

The algorithms (GA, SAAVG, SAMCP, SAS) were used to find minimal values (f_{min}) of five benchmark, multimodal functions : f_1 – Schwefel’s function, f_2 – Griewangk’s function, f_3 – Ackley's Path function, f_4 – Branins’s function, f_5 – Himmelblau’s function. The functions we employed are listed in Table 1.

One fitness function was used and based on commonly known mean absolute error measure (MAE) expressed in the form of formula (3) where y_i stands for the actual value and \hat{y}_i – predicted value of i-th case. The fitness function, denoted by MAE_y,

was calculated for the output value of a given benchmark function. It determined how near the optimum was the output value of the function. In this case N was always equal 1.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \tag{3}$$

All four algorithms: GA, SAAVG, SAMCP, SAS, were executed independently 50 times and the final values of MAE_y were calculated as an average over 50 runs for best individuals found by respective algorithms. 50 initial populations composed of 100 chromosomes were randomly created and they were the same for all algorithms in each run. In order to investigate the convergence of individual algorithms, 1000 generations were carried out in each run. The values of MAE_y were calculated for each algorithm.

Moreover, nonparametric Friedman and Wilcoxon signed-rank tests were carried out for MAE_y provided by the algorithms by the 1000-th generation over 50 independent runs for individual benchmark functions.

4 Results of Experiments

The performance of GA, SAAVG, SAMCP, SAS algorithms on respective benchmark functions in respect of MAE_y measures was shown in Fig. 7-11.

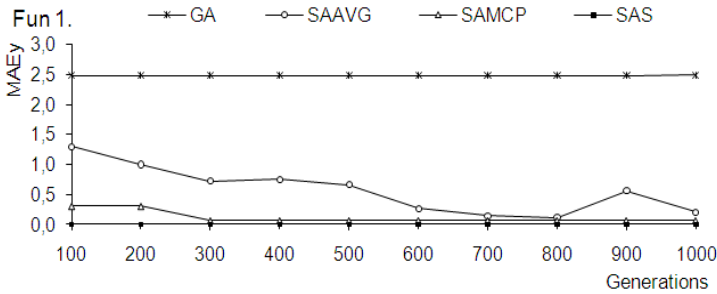


Fig. 7. Performance of algorithms on Schwefel's function in terms of MAE_y

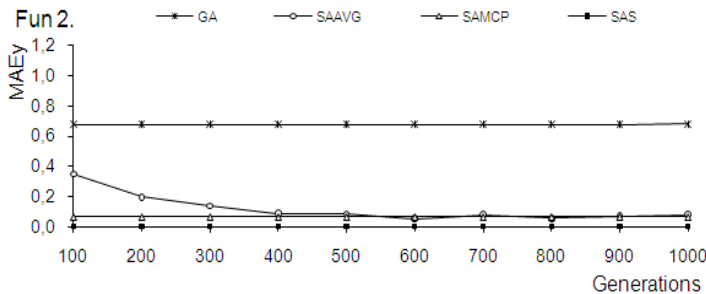


Fig. 8. Performance of algorithms on Griewangk's function in terms of MAE_y

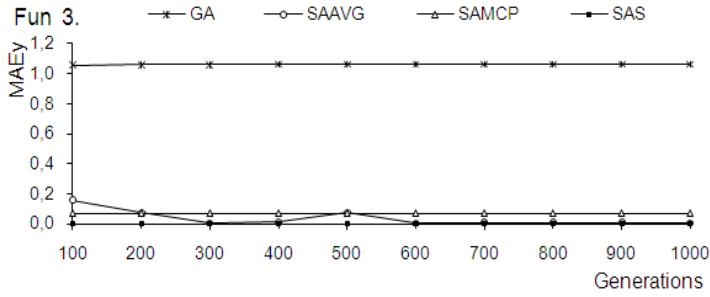


Fig. 9. Performance of algorithms on Ackley's Path function in terms of MAE_y

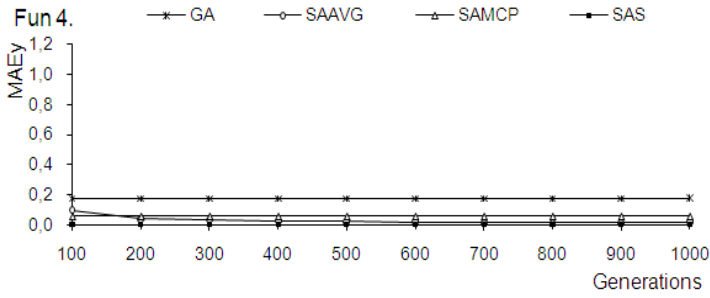


Fig. 10. Performance of algorithms on Rastrigin's function in terms of MAE_y

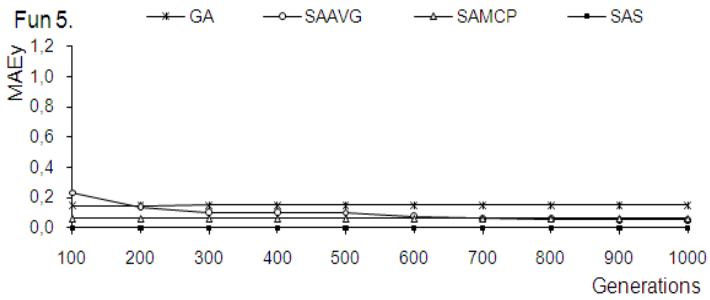


Fig. 11. Performance of algorithms on Griewangk's function in terms of MAE_y

Table 2. Average rank positions of algorithms determined during Friedman test

	1st	2nd	3rd	4th
f_1	SAS (1.06)	SAMCP (2.47)	SAAVG (2.83)	GA (3.64)
f_2	SAS (1.07)	SAAVG (2.65)	SAMCP (2.69)	GA (3.59)
f_3	SAAVG (1.99)	SAS (2.33)	SAMCP (2.76)	GA (2.92)
f_4	SAS (1.01)	SAAVG (2.67)	SAMCP (2.77)	GA (3.55)
f_5	SAS (1.00)	SAMCP (2.64)	SAAVG (3.15)	GA (2.21)

In each case SAS, SAMCP, SAAVG revealed better convergence than GA. Moreover, SAS, SAAVG and SAMCP produced lower values of MAEy than GA. SAS achieved the best results for all functions and in each case. The advantage of self-adapting algorithms over GA is apparent particularly on Schwefel's, Griewangk's, Ackley's Path function.

The Friedman test performed in respect of MAEy values of all algorithms by the 1000-th generation over 50 independent runs for individual benchmark functions showed that there are significant differences between some algorithms. Average ranks of individual algorithms are shown in Table 2, where the lower rank value the better algorithm. The results of the Wilcoxon test are given in Table 3, where +, -, and \approx denote that the first algorithm in a pair performed significantly better than, significantly worse than, or statistically equivalent to the second algorithm, respectively. Main outcome is as follows: GA was significantly worse than SAS, SAAVG, SAMCP for each benchmark function, besides one case. SAS was significantly better than any other algorithm for each benchmark function, besides one case. Differences between SAAVG and SAMCP are not so clear, the most frequently observation is as follows: there are is statistically significant difference between them.

Table 3. Results of Wilcoxon tests for SAS, SAAVG, SAMCP, and GA algorithms

Alg vs Alg	f_1	f_2	f_3	f_4	f_5
SAS vs SAAVG	+	+	\approx	+	+
SAS vs SAMCP	+	+	+	+	+
SAS vs GA	+	+	+	+	+
SAAVG vs SAMCP	\approx	\approx	+	\approx	-
SAAVG vs GA	+	+	+	+	\approx
SAMCP vs GA	+	+	+	+	+

5 Conclusions and Future Work

The experiments aimed to compare the convergence of a classic genetic algorithm (GA) with three self-adapting genetic algorithms (SAAVG, SAMCP, SAS) in which the rates of mutation and crossover, number of crossover points, number of genes to mutation or population size were dynamically evolved. Five multimodal, benchmark functions were employed.

The results showed that almost all self-adaptive algorithms revealed better convergence than the traditional GA. SAS was the best algorithm for all functions, except one case, for function 3, where is no significant differences between SAS and SAAVG. The advantage of all proposed self-adapting algorithms over GA became particularly apparent on Schwefel's, Griewangk's and Ackley's Path function.

Statistical nonparametric Friedman and Wilcoxon signed-rank tests allowed for the analysis of behaviour of the algorithms on individual benchmark functions.

Further research is planned where other criteria of the algorithm assessment will be taken into account. The possible application of the self-adaptive techniques first, to

create ensemble models and the second, to create genetic fuzzy models assisting with real estate appraisal will also be considered.

Acknowledgments. This work was funded partially by the Polish National Science Centre under grant no. N N516 483840.

References

1. Angeline, P.J.: Adaptive and self-adaptive evolutionary computations. In: Palaniswami, M., Attikiouzel, Y. (eds.) *Computational Intelligence: A Dynamic Systems Perspective*, pp. 152–163. IEEE Press, New York (1995)
2. Bäck, T.: Self-adaptation in genetic algorithms. In: Varela, F.J., Bourgine, P. (eds.) *Proc. First European Conference on Artificial Life, Toward a Practice of Autonomous Systems*, pp. 263–271. MIT Press, Cambridge (1992)
3. Bäck, T., Schwefel, H.-P.: An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation* 1(1), 1–23 (1993)
4. Cervantes, J., Stephens, C.S.: Limitations of Existing Mutation Rate Heuristics and How a Rank GA Overcomes Them. *IEEE Transactions on Evolutionary Computation* 13(2), 369–397 (2009)
5. Deb, K., Beyer, H.-G.: Self-adaptive genetic algorithms with simulated binary crossover. *Evolutionary Computation* 9(2), 197–221 (2001)
6. Digalakis, J.G., Margaritis, K.G.: An Experimental Study of Benchmarking Functions for Genetic Algorithms. *Int. J. Computer Math.* 79(4), 403–416 (2002)
7. Eiben, E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 3(2), 124–141 (1999)
8. Herrera, F., Lozano, M.: Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions. *Soft Computing* 7(8), 545–562 (2003)
9. Hinterding, R., Michalewicz, Z., Eiben, A.E.: Adaptation in Evolutionary Computation: A Survey. In: *Proceedings of the Fourth International Conference on Evolutionary Computation (ICEC 1997)*, pp. 65–69. IEEE Press, New York (1997)
10. Król, D., Lasota, T., Trawiński, B., Trawiński, K.: Investigation of evolutionary optimization methods of TSK fuzzy model for real estate appraisal. *International Journal of Hybrid Intelligent Systems* 5(3), 111–128 (2008)
11. Lasota, T., Trawiński, B., Trawiński, K.: Evolutionary Generation of Rule Base in TSK Fuzzy Model for Real Estate Appraisal. In: *Proc. 3rd Int. Workshop on Genetic and Evolving Fuzzy Systems (GEFS 2008)*, pp. 71–76. IEEE Press, New York (2008)
12. Maruo, M.H., Lopes, H.S., Delgado, M.R.: Self-Adapting Evolutionary Parameters: Encoding Aspects for Combinatorial Optimization Problems. In: Raidl, G.R., Gottlieb, J. (eds.) *EvoCOP 2005*. LNCS, vol. 3448, pp. 154–165. Springer, Heidelberg (2005)
13. Meyer-Nieberg, S., Beyer, H.-G.: Self-Adaptation in Evolutionary Algorithms. In: Lobo, F.G., Lima, C.F., Michalewicz, Z. (eds.) *SCI*, vol. 54, pp. 47–75. Springer, Heidelberg (2007)
14. Pohlheim, H.: GEATbx Examples. Documentation for GEATbx version 3.7 (2005)
15. Schaffer, J.D., Morishima, A.: An adaptive crossover distribution mechanism for genetic algorithms. In: *Genetic Algorithms and their Application: Proceedings of the Second International Conference on Genetic Algorithms*, pp. 36–40 (1987)

16. Smętek, M., Trawiński, B.: Investigation of Genetic Algorithms with Self-adaptive Crossover, Mutation, and Selection. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS (LNAI), vol. 6678, pp. 116–123. Springer, Heidelberg (2011)
17. Smith, J.E., Fogarty, T.C.: Operator and parameter adaptation in genetic algorithms. *Soft Computing* 1(2), 81–87 (1997)
18. Tang, K., Li, X., Suganthan, P.N., Yang, Z., Weise, T.: Benchmark Functions for the CEC 2010 Special Session and Competition on Large Scale Global Optimization, Technical Report, Nature Inspired Computation and Applications Laboratory, USTC, China (2009), <http://nical.ustc.edu.cn/cec10ss.php>
19. Yao, X., Liu, Y.: Fast evolution strategies. *Contr. Cybern.* 26(3), 467–496 (1997)
20. Yao, X., Liu, Y., Lin, G.: Evolutionary Programming Made Faster. *IEEE Transactions on Evolutionary Computation* 3(2), 91–96 (1999)

Multiobjective Particle Swarm Optimization Using Fuzzy Logic

Hossein Yazdani¹, Halina Kwasnicka¹, and Daniel Ortiz-Arroyo²

¹ Institute of Informatics, Wrocław University of Technology, Wrocław, Poland

² Electronics Department, Computational Intelligence and Security Laboratory,
Aalborg University, Denmark

hyazda10@student.aau.dk, halina.kwasnicka@pwr.wroc.pl, do@es.aau.dk

Abstract. The paper presents FMOPSO a multiobjective optimization method that uses a Particle Swarm Optimization algorithm enhanced with a Fuzzy Logic-based controller. Our implementation makes use of a number of fuzzy rules as well as dynamic membership functions to evaluate search spaces at each iteration. The method works based on Pareto dominance and was tested using standard benchmark data sets. Our results show that the proposed method is competitive with other approaches reported in the literature.

Keywords: Particle Swarm Optimization, Fuzzy Logic, multiobjective optimization.

1 Introduction

Optimization modeling is one of the most powerful techniques to find optimal solutions of problems in application areas such as economy, industry, finance, and others. For instance in economic applications, profit and sales must be maximized and cost should be as low as possible [5]. The goal is to find the best solution x^* from a set of possible solutions X according to a set of criteria $F = \{f_1, f_2, \dots, f_n\}$. This set of criteria is expressed as a mathematical function named objective function [5].

General optimization (GO) methods are categorized into two main classes: deterministic and probabilistic methods. The deterministic methods work based on heuristics, such as adding punishment to escape from local minima. Heuristics use the information currently gathered by the algorithm to help deciding which solution candidate should be tested next or how the next solution can be produced [5]. Probabilistic methods estimate probabilities to decide whether the search should depart from the neighbourhood of a local minimum.

Particle Swarm Optimization is one of the methods that have been proposed to solve GO problems [6].

Multiobjective optimization algorithms are designed to optimize a set of objective functions. The simplest methods rely on optimizing the weighted sum $g(x)$ of all functions (criteria) $f_i(x) \in F$ [5]. The mathematical foundations for multiobjective optimization that considers conflicting criteria in a fair way was

laid by Vilfredo Pareto 110 years ago [5]. Pareto optimization is based on the definition of domination: a solution x_1 dominates (is preferred to) solution x_2 ($(x_1 \vdash x_2)$) if x_1 is better than x_2 in at least one objective function and not worse with respect to all other objectives.

A solution $x^* \in X$ is Pareto optimal (belongs to the optimal set X^*) if it is not dominated by any other solution in the problem space X . In terms of Pareto optimization, X^* is called the Pareto optimal set, denoted as P^* [6].

The set $PF^* = \{f_1(x), f_2(x), \dots, f_k(x) \mid x \in P^*\}$ is called Pareto Front PF^*

Particle Swarm Optimization (PSO) was introduced by R.C. Eberhart and J. Kennedy in 1995 [6]. PSO is an adaptive, global optimization method which is based on updating the value of each particle to obtain the best solution. In this method, each potential solution is called a *particle*, and each particle has a *fitness value* which is calculated by a fitness function. This fitness value is the one that should be optimized. PSO works by generating some random solutions consisting of particles, where each particle is iteratively updated with two different values. One is the best value reached by the particle so far (called *local best* or *lbest*), and the second one is the best value obtained by any particle so far (called *global best* or *gbest*) [7]. Based on these two best values, the velocity of particle p and its position are updated and then the fitness of the particle is calculated. In this way an optimal solution may be found after some number of iterations. There are two different general PSO models, one is called *local version* and another is *global version* [6]. In the local version, each particle flies through the search space to adjust the velocity according to its best achieved performance so far and the best performance achieved by the neighbourhood particles. In the global version, the particle's velocity is adjusted according to its best achieved performance so far and the best performance achieved by all particles [2].

The main idea of the algorithm presented in this paper is to break down the problem to be solved into several simpler ones, and evolve particles to find Pareto Fronts in smaller spaces. A combination of these fronts constitutes the final Pareto Front. Fuzzy logic is used in deciding which part of the problem should be selected for the next iteration. The objective is to use that part, on which finding the non-dominated particles is more probable.

This paper is organized in the following way. In section 2 some related work is briefly described. The proposed approach is presented in section 3. The experimental study of the method is described in section 4. The last section summarizes the paper and presents some conclusions.

2 Related Work

PSO has become a popular optimization method that is widely studied and compared to other approaches. Given that the literature on this subject is extensive, in this section we present a brief summary of related work.

[3] describes the use of PSO algorithm as a tool to optimize path finding. Authors use the Mamdani inference fuzzy system to decide when to increase or decrease the velocity of particles and in which direction the positions of particles should be changed.

A new algorithm was introduced in [1] where the search space is divided according to the best achieved values accumulated in a repository. The algorithm works based on a hypercube. Then from the chosen hypercube (on the basis of fitness of all hypercubes) one particle is selected randomly.

POS is combined with Fuzzy Logic in [4], where Fuzzy Logic helps the particle to improve an existing solution by replacing the low quality links with high quality links. The local version of PSO model is used to optimize topology design of distributed local area networks.

PSO and multiobjective optimization (MO) in n -dimensional search space are discussed in [6]. Authors discuss weighted aggregation approaches such as: Dynamic weighted aggregation (DWA), Conventional weighted aggregation (CWA), Bang-Bang Weighted Aggregation (BWA), Vector Evaluated Genetic Algorithm (VEGA) and Vector Evaluated Particle Swarm Optimization (VEPSO). Authors introduced a maximum value for velocity to improve the performance of the basic PSO by avoiding high increases in velocity values. Another topic discussed in that paper is the range of values for the basic parameters c_1, c_2 (connected with the influence of local and global best solution).

In [8], a new algorithm for two-objective functions in two-dimensional fitness value space is introduced. Authors assume fixed fitness values of the first objective function, and try to optimize the second objective function.

Concepts of PSO, neighbourhood topology, multi-objective optimization (MOO) and leaders in MOO are discussed in [12]. Authors explain how to select the leader from all non-dominated solutions as a global best solution to guide the algorithm to get the new particles. Nearest neighbor density estimator and kernel density estimator are some approaches mentioned in this paper.

3 Multiobjective Particle Swarm Optimization Using Fuzzy Logic (FMOPSO)

The FMOPSO method is inspired by the "Divide and conquer" or "Sub population" approach. The main idea of FMOPSO is to break the original problem down to several parts with less complexity. Then by gathering particles (solutions) from these smaller problems, a better overall solution could be obtained. The search space is divided into an arbitrary number of major spaces (we use five major spaces), further, every major space is divided into an arbitrary number of minor spaces (in our experiments it is also five). To obtain better results, we combined population and pareto-based approaches. Our algorithm determines which area has more priority for being selected as a new search space for next iteration. For this purpose we make use of a fuzzy controller that evaluates search spaces at each iteration. In the calculation of velocity and position of a new particle we use the basic PSO algorithm:

$$\begin{aligned} v_{(i+1)} &= v_i + c_1 * r_1 * (lbest_i - p_i) + c_2 * r_2 * (gbest_i - p_i) \\ p_{(i+1)} &= p_i + v_{(i+1)} \end{aligned} \quad (1)$$

where $c_1 = 1.4$, $c_2 = 1.5$, r_1, r_2 are random value in $[0, 1]$. Half a range of the variable is the limit for maximal velocity. When a particle's velocity is near to zero, the particle is moved to a different dimension in the n -dimensional search space. This allows particles to escape from the local optimum. Additionally, we use two accumulators, named as *repository* and *deleted-repository*, thanks to it the fuzzy controller can check densities of both – non-dominated and dominated particles. The fuzzy controller decides which search space should be selected to choose the *gbest*.

The main steps of FMOPSO. The FMOPSO algorithm is briefly presented in Algorithm 1. The first step, *Initialize parameters*, consists of four methods, *Get ranges*, *Initialize particles*, *Initialize fitness*, and *Initialize parts*, i.e., major and minor parts.

In *Compute velocity* we perform three steps, each has a different strategy for choosing *gbest*: (1) select *gbest* from all *gbest* achieved so far (if we have many candidates, one is randomly selected), (2) fuzzy controller looks for the best search space to get *gbest* with respect to the density of particles in both repository and deleted-repository, (3) look for the *gbest* based on the major part. *Evaluate fitness* evaluates the particle's fitness value (the objective functions). *Update Lbest* method updates best position achieved by the considered particle so far. *Evaluate Non-dominated* method is used to choose the non-dominated particle.

Algorithm 1. FMOPSO general algorithm

```

Initialize parameters();
Update Lbest();
Evaluate Non-dominated();
while  $i \leq \text{max.iteration}$  do
    Compute Velocity();
    Evaluate fitness();
    Update Lbest();
    Evaluate Non-dominated();
     $i = i + 1$ ;
end while
Make report();

```

Fuzzy Logic Controller. Whole search space is divided into five smaller parts, and every smaller search subspace is divided into five subsections. The task is to decide which subsection should be selected for next iteration. This is done on the basis of evaluating the availability of non-dominated particles in a search space. Fuzzy controller checks the density of particles in both the repository and deleted-repository. The interesting area is where the density of dominated particle is low or medium and the density of non-dominated particle is medium and high. This is the reason why our fuzzy controller looks at densities in both repositories.

Membership Functions for Input and Output Variables. Our fuzzy controller uses three fuzzy sets for each of the input variables, *Density* and *No. of*

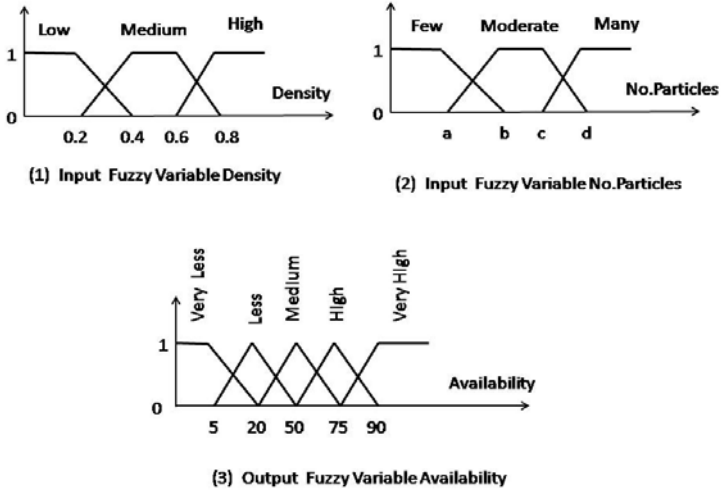


Fig. 1. Graphical representation of membership function

Particles, for both accumulators – repository and deleted-repository. For variable *Density* they are: *Low* (L), *Medium* (M), and *High* (H). For *No. Particles*: *Few* (F), *Moderate* (MO), and *Many* (MA).

Five fuzzy sets are defined for output variable *Availability* of non-dominated particles: *Very High Availability* (VHP), *High Availability* (HP), *Medium Availability* (MP), *Less Availability* (LSP), *Very Less Availability* (VLSP).

Membership functions are presented in Fig. 1. It is seen in this figure, that the boundaries of fuzzy sets for variable No. of Particles change with increasing iterations. In that figure the values of points *a*, *b*, *c*, *d* in the fuzzy sets are calculated using a coefficient $coefficient = 0.1 * population.size * current.iteration$, in the following way:

$$\begin{aligned}
 a &= coefficient, \\
 b &= coefficient + coefficient/2, \\
 c &= 2 * coefficient + coefficient/2, \\
 d &= 3 * coefficient.
 \end{aligned}$$

Fuzzy Rule Base. We have defined eighteen different fuzzy rules for the controller. They are shown in Table 1. The detailed rules are:

1. If *density* in dominated set is low or medium and *density* in non-dominated set is low and *NO.Particles* is few, *availability* is high
2. If *density* in dominated set is low or medium and *density* in non-dominated set is low and *NO.Particles* is moderate, *availability* is middle
3. If *density* in dominated set is low or medium and *density* in non-dominated set is low and *NO.Particles* is many, *availability* is less
4. If *density* in dominated set is high and *density* in non-dominated set is low and *NO.Particles* is few, *availability* is middle

Table 1. 3-dimensional matrix defining the fuzzy rules: availability depending on densities in the both accumulators

Repository →		L			M			H		
Del-rep ↓	No.P →	F	MO	MA	F	MO	MA	F	MO	MA
L		HP	MP	LSP	HP	HP	VHP	HP	VHP	VHP
M		HP	MP	LSP	MP	MP	MP	MP	HP	HP
H		MP	LSP	VLSP	MP	LS	VLSP	MP	MP	LSP

5. If *density* in dominated set is high and *density* in non-dominated set is low and *NO.Particles* is moderate, *availability* is less
6. If *density* in dominated set is high and *density* in non-dominated set is low and *NO.Particles* is moderate, *availability* is very Less
7. If *density* in dominated set is low and *density* in non-dominated set is medium and *NO.Particles* is few or moderate, *availability* is very high
8. If *density* in dominated set is low and *density* in non-dominated set is medium and *NO.Particles* is many, *availability* is very high
9. If *density* in dominated set is medium and *density* in non-dominated set is medium and *NO.Particles* is few or moderate or much, *availability* is middle
10. If *density* in dominated set is high and *density* in non-dominated set is medium and *NO.Particles* is few, *availability* is middle
11. If *density* in dominated set is high and *density* in non-dominated set is medium and *NO.Particles* is moderate, *availability* is less
12. If *density* in dominated set is high and *density* in non-dominated set is medium and *NO.Particles* is many, *availability* is very less
13. If *density* in dominated set is low and *density* in non-dominated set is high and *NO.Particles* is few, *availability* is high
14. If *density* in dominated set is low and *density* in non-dominated set is high and *NO.Particles* is moderate or many, *availability* is very high
15. If *density* in dominated set is medium and *density* in non-dominated set is high and *NO.Particles* is few, *availability* is middle
16. If *density* in dominated set is medium and *density* in non-dominated set is high and *NO.Particles* is moderate or many, *availability* is high
17. If *density* in dominated set is high and *density* in non-dominated set is high and *NO.Particles* is few or moderate, *availability* is very middle
18. If *density* in dominated set is high and *density* in non-dominated set is high and *NO.Particles* is many, *availability* is less

The fuzzy controller fires the applicable rules and calculates the fuzzy output using Mamdani (max-min) implication technique. To get the crisp value from all fired rules, the weighted-average defuzzification technique is used.

4 Experimental Results

To verify the performance of our method, FMOPSO was run using different benchmark data sets and different values of parameters: five test functions (ZDT test set), 100 iterations, and for five different numbers of particles (5, 25, 50, 75, 100).

Table 2. ZDT two-objectives problems

Name	Problem	Type	Parameter Domains
ZDT1	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$	Convex	[0,1]
ZDT2	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(f_1, g) = 1 - (\frac{f_1}{g})^2$	Non-Convex	[0,1]
ZDT3	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} - (\frac{f_1}{g}) \sin(10\pi f_1)$	Non-Convex, Dis-connected	[0,1]
ZDT4	$f_1(x) = x_1$ $g(x) = 1 + 10(n-1) + \sum_i i = 2^n(x_i^2 - 10 \cos(4\pi x_i))$ $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$	Convex, Multi-Modal	[0,1]
ZDT6	$f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $g(x) = 1 + 9(\frac{\sum_{i=2}^{10} x_i}{9})^{0.25}$ $h(f_1, g) = 1 - (\frac{f_1}{g})^2$	Non-Convex, Non-uniformly Spaced	[0,1]

Zitzler-Deb-Thiele (ZDT) set contains scalable problems according to the number of decision variables [9,10,11]. ZDT problems contain two objective problems. Given f_1 , the second criterion is a composite function $f_2(x) = g(x)h(f_1(x), g(x))$, both objectives should be minimized. Table 2 contains defined 2-dimensional problems. Three performance metrics were used: Generational distance, Spacing and Error ratio.

Generational Distance (GD). GD was introduced by Van Veldhuizen and Lamont [12] to calculate the distance between particles in non-dominated set generated by the method and the Pareto Optimal set.

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n}, \quad d_i = \min_{k=1}^{|p^*|} \sqrt{\sum_{m=1}^M (f_m^i - f_m^{*(k)})^2} \quad (2)$$

where n is the number of particles in the non-dominated set and d_i is the Euclidean distance between particle in the non-dominated set and the closest one from Pareto Optimal set and $f_m^{*(k)}$ is the m -th objective value of the k -th member of p^* . It is obvious that the smaller values of GD are preferred. GD close to zero indicates that non-dominated set is a part of Pareto Front.

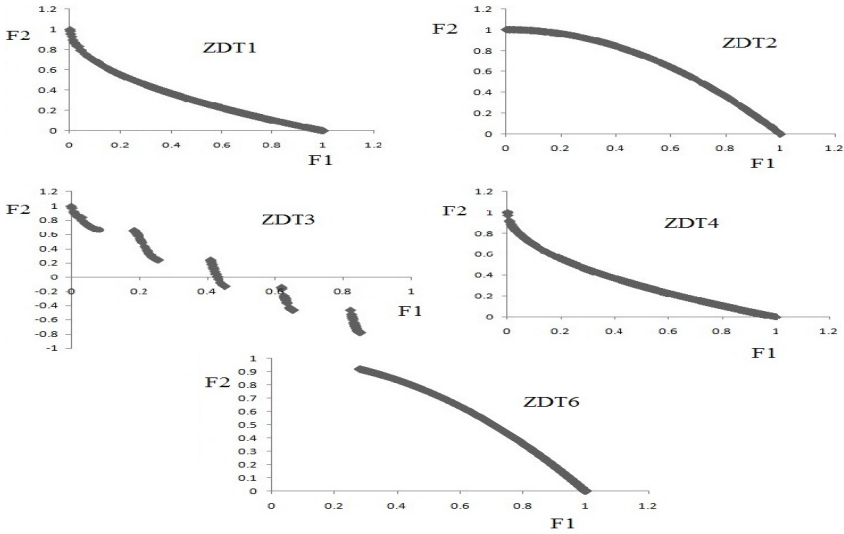


Fig. 2. Pareto front produced by FMOPSO for the ZDT set test problem

Spacing (SP). SP was introduced by Schott(1995) [12] to calculate distance between consecutive solutions from non-dominated set.

$$S = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - \bar{d})^2} \quad \bar{d} = \sum_{i=1}^n \frac{d_i}{n} \tag{3}$$

The smaller spacing values are better, they are the standard deviations of different d_i values.

Error Ratio (ER). ER was introduced by Van Velshuizen [12] to measure the number of non-dominated particles which are not the member of Pareto Optimal set.

$$ER = \frac{\sum_{i=1}^n e_i}{n} \tag{4}$$

where $e_i = 1$ if i is not a member of Pareto Optimal set and $e_i = 0$ otherwise. $ER = 0$ indicates that all solutions in non-dominated set are members of Pareto Optimal set. Figure 2 presents the results produced by our method.

Table 3 shows the result of FMOPSO compared to common Evolutionary Algorithms: NSGA-II real coded, NSGA-II binary coded, SPEA and PAES. In this table the results displayed correspond to the mean and variance with respect to the convergence metric.

Table 4 shows the result achieved by FMOPSO and other recently presented MOPSO algorithms: OMPSO, SMPSO, MOPSO-TVAC, MOPSO-TVIW. In this table the results displayed correspond to the median and IQR indicator with respect to the Delta metric.

Table 3. Convergence metric

Algorithm		ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
NSGA-II Real-coded	$\bar{\gamma}$	0.033482	0.072391	0.114500	0.513053	0.296564
	δ_γ^2	0.004750	0.031689	0.007940	0.118460	0.013135
NSGA-II Binary-coded	$\bar{\gamma}$	0.000894	0.000824	0.043411	3.227636	7.806798
	δ_γ^2	0	0	0.000042	7.307630	0.001667
SPEA	$\bar{\gamma}$	0.001249	0.003043	0.044212	9.513615	0.020166
	δ_γ^2	0	0.000020	0.000019	11.321067	0.000923
PAES	$\bar{\gamma}$	0.082085	0.126276	0.023872	0.854816	0.085469
	δ_γ^2	0.008679	0.036877	0.00001	0.527238	0.006664
FMOPSO	$\bar{\gamma}$	0.00028	0.00000287	0.00060	0.00004	0.00016
	δ_γ^2	0.00000	0.00000	0.00000	0.00000	0.00000012

Table 4. Delta metric

Algorithm		ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
OMOPSO	\bar{x}	$7.98e^{-02}$	$7.46e^{-02}$	$7.13e^{-01}$	$8.69e^{-01}$	$2.90e^{-01}$
	IQR	$1.4e^{-02}$	$1.6e^{-02}$	$1.0e^{-02}$	$5.9e^{-02}$	$1.1e^{+00}$
SMPSO	\bar{x}	$7.66e^{-02}$	$7.33e^{-02}$	$7.10e^{-01}$	$9.81e^{-02}$	$2.83e^{-01}$
	IQR	$1.4e^{-02}$	$1.6e^{-02}$	$7.2e^{-03}$	$1.4e^{-02}$	$1.2e^{+00}$
MOPSO-TVAC	\bar{x}	$1.01e^{-01}$	$8.71e^{-01}$	$7.81e^{-01}$	$2.05e^{-01}$	$1.33e^{+00}$
	IQR	$1.3e^{-02}$	$1.3e^{-02}$	$7.1e^{-02}$	$3.6e^{-02}$	$5.7e^{-02}$
MOHPSO	\bar{x}	$1.10e^{-01}$	$9.01e^{-02}$	$7.71e^{-01}$	$9.02e^{-01}$	$1.29e^{+00}$
	IQR	$2.7e^{-02}$	$1.9e^{-02}$	$5.9e^{-02}$	$1.6e^{-01}$	$4.3e^{-02}$
MOPSO-TVIW	\bar{x}	$8.39e^{-02}$	$7.09e^{-02}$	$7.12e^{-01}$	$1.29e^{-01}$	$1.11e^{+00}$
	IQR	$1.6e^{-02}$	$2.0e^{-02}$	$9.5e^{-03}$	$3.4e^{-01}$	$1.2e^{+00}$
FMOPSO	\bar{x}	$1.75e^{-02}$	$1.75e^{-02}$	$4.0e^{-02}$	$1.9e^{-02}$	$1.5e^{-3}$
	IQR	$3.88e^{-02}$	$1.2e^{-03}$	$5.1e^{-03}$	$4.6e^{-04}$	$4.1e^{-04}$

5 Conclusions and Future Work

We have proposed a new algorithm that combines a population based optimization techniques, the Pareto based approach, and a fuzzy controller. The experiments with the ZDT benchmark data sets indicate that FMOPSO approach is able to find the non-dominated particles that are close to the Pareto Front.

Our modification of PSO lies in adding some fuzzy knowledge to make the method more intelligent. Thanks to this knowledge FMOPSO decides where the best area of search space is.

The complexity of the method could be reduced by using sorted balanced tree (AVL or R & B tree) instead of the current array that is being used. The complexity of updating repository is currently $O(kN^2)$ where N is the size of swarm and k is the number of objectives. Complexity of the updating process for all iterations (M) is $O(kMN^2)$ [12], where complexity of insertion into AVL tree is at most $O(\log(N + \frac{3}{2} + \log(\sqrt{5}) - 3))$ rebalancing operations and $O(\log(N +$

$\frac{3}{2} + \log(\sqrt{5}) - 4$ rebalancing operations for deletion, if N is the maximum size of nodes [13]. The use of this kind of tree will improve the time for updating the repository.

The approach used in FMOPSO, namely – finding the most promising areas of search space, i.e., the suitable ranges of particular variables, seems to be good way for multi-objective optimization methods. For example, the best values of variables x_2, x_3, \dots, x_{30} from the range $[0,1]$ in ZDT1 data is zero. The system should find the best area for these variables by analyzing them with respect to the given ranges. Searching the promising subspaces at each iteration, on the basis of history analysis, should allow the method to find very good solutions in relatively short time. In future work we plan to improve the fuzzy controller to include a user knowledge base and other fuzzy sets.

References

1. Weise, T.: Global Optimization Algorithms Theory and Application. EBook. IEEE Press, Los Alamitos (2009)
2. Parsopoulos, K.E., Vrahatis, M.N.: Recent approaches to global optimization problems through Particle Swarm Optimization. *Natural Computing* 1, 235–306 (2002)
3. Clerc, M.: Particle Swarm Optimization. Wiley-ISTE, Chichester (2006)
4. Das, S., Abraham, A., Konar, A.: Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives. *Studies in Computational Intelligence (SCI)*, vol. 116, pp. 1–38 (2008)
5. Ghanizadeh, A., Sinaie, S., Abarghouei, A.A., Shamsuddin, S.M.: A fuzzy-particle swarm optimization based algorithm for solving shortest path problem. In: 2nd International Conference on Computer Engineering and Technology, ICCTET, pp. V6-404–V6-408 (2010)
6. Coello Coello, C.A., Lechuga, M.S.: MOPSO: a proposal for multiple objective particle swarm optimization. In: *Proceedings of the 2002 Congress on Evolutionary Computation*, vol. 2, pp. 1051–1056 (2002)
7. Khan, S.A.: Design and analysis of evolutionary and swarm intelligence techniques for topology design of distributed local area networks, ch. 9, University of Pretoria (2009)
8. Hu, X., Eberhart, R.: Multiobjective optimization using dynamic neighborhood particle swarm optimization. In: *Congress on Evolutionary Computation*, vol. 2, pp. 1677–1681 (2002), 0-7803-7282-4/02 IEEE
9. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* 10(5), 477–506 (2006)
10. Coello Coello, C.A., Dhaenens, C., Jourdan, L.: *Advances in Multi-Objective Nature Inspired Computing*. Springer, Heidelberg (2010)
11. 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)
12. Reyes-Sierra, M., Coello Coello, C.A.: Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research* 2(3), 287–308 (2006)
13. Larsen, K.S.: AVL trees with relaxed balance. In: *Parallel Processing Symposium*, pp. 888–893 (1994)

An Evolutionary Algorithm for the Urban Public Transportation

Jolanta Koszelew

Faculty of Computer Science, Bialystok University of Technology
j.koszelew@pb.edu.pl

Abstract. A public transport route planner provides for citizens and tourists information about available public transport journeys. The heart of such systems are effective methods for solving itinerary planning problem in a multi-modal urban public transportation networks. This paper describes a new evolutionary algorithm solving a certain version of this problem. The method returns the set of k -journeys that lexicographically optimize two criteria's: total travel time and number of transfers. Proposed algorithm was compared with two another solutions for itinerary planning problem. This comparison is prepared on the base of experimental results which were performed on real-life data - Warsaw city public transport network. Conducted experiments confirm high effectiveness of the proposed method in comparison with two another known solutions for considered problem.

Keywords: multi-modal public transport networks, itinerary planning problem, time-dependent k -shortest paths problem, evolutionary algorithm.

1 Introduction

A journey in a modern urban public transport network usually involves combined use of the available public transport services. Any path in such journey enhanced with a feasible schedule to traverse it is called itinerary. The itinerary planning problem in a multi-modal urban public transport network consists of finding optimal journey or set of journeys satisfying user's preferences. Generally, the itinerary planning problem constitutes a multi-criteria time-dependent routing and planning problem [8], providing a user with many alternative itineraries for a given urban journey. A public transportation journey planner is a kind of Intelligent Transportation Systems (ITS) and provides information about available public transport journeys. Users of such a system determine source and destination point of the travel, the start time, their preferences and, as a result the system returns information about optimal routes (journeys). In practice, public transport users' preferences may be various, but the most important of them are: a minimal travel time and a minimal number of changes (from one vehicle to another).

The shortest path problem is a core model that lies at the heart of network optimization. It assumes that weight link in traditional network is static, but

is not true in many fields of ITS. The optimal path problems in variable-time network break through the limit of traditional shortest path problems and become foundation theory in ITS. The new real problems make the optimal path computing to be more difficult than finding the shortest paths in networks with static and deterministic links, meanwhile algorithms for a scheduled transportation network are time-dependent.

Many algorithms has been developed for networks whose edge weights are not static but change with time but most of them take into consideration a network with only one kind of link, without parallel links and returns only one route. Cooke and Halsey [4] modified Bellman's [2] "single-source with possibly negative weights" algorithm to find the shortest path between any two vertices in a time-dependent network. Dreyfus [5] made a modification to the standard Dijkstra algorithm to cope with the time-dependent shortest path problem. Orda and Rom [12] discussed how to convert the cost of discrete and continuous time networks into a simpler model and still used traditional shortest path algorithms for the time-dependent networks. Chabini [3] presented an algorithm for the problem with discrete time and edge weights are time-dependent. Ahuja [1] proved that finding the general minimum cost path in a time-dependent network is NP-hard and special approximation method must be used to solve this problem.

The evolutionary algorithm (EA) presented in this paper generates k routes with optimal travel time and number of transfers in lexicographically order. Narrowly, resultant routes have the minimal travel time in first order and the minimal number of transfers in second order. Like the k -shortest paths algorithm [11], these methods generate multiple "better" paths, the user can have more choices from where he or she can select according own preferences such as total amount of fares, convenience, preferred routes and so on. Presented algorithm is an improved version of the method described in [10]. Other EA approach to Urban Public Transportation Problem we can find in [14] but in this version of problem only one resultant route is returned.

The computational performance of EA had been tested on a wide range of real-life journey planning problems defined on the urban public transport network of Warsaw, Poland. EA was also compared with two another methods solving the itinerary planning problem. The first comparable algorithm realizes an cultural algorithm (CA) [13]. CAs are an evolutionary computation technique, that uses knowledge that has been generated in several times, for the same population, using a belief space. In the belief space CA stores routes with a minimal number of transfers and in each iteration of the method special operators try to improved a realization time of routes included in the population. The second method applies a local search heuristic based on a special transfer graph (TG) [6]. In this algorithm routes with smallest number of transfers are considered. Next, TG tries to add new stops to such routes, but insertion is performed only if the realization time of modified route doesn't increased. Computer experiments had shown that EA generates routes as good as CA, better than TG and is significantly faster than CA.

The remainder of this paper consists of five sections. Section 2 includes definition of itinerary planning problem and description of multi-modal urban public transport network model. In Section 3 author in detail presents each step of EA and illustrates it with a simple example. Section 4 is the comparison of effectiveness of EA, CA and TG methods in two aspects: computation time and quality of resultant journeys. The paper ends the section which also includes the major concluding remark.

2 Network Model and Problem Definition

A public transportation network in our model is represented as a bimodal weighted graph $G = \langle V, S, W \rangle$ [9], where V is a set of nodes, S is a set of transport links and W is a set of walk links. Each node in G corresponds to a certain transport stop (bus, tram or metro stop, etc.), shortly named stop. We assume that stops are represented with numbers from 1 to n . The directed edge (i, j, l, t) is an element of the set S , if the line number l connects the stop number i as a source point and the stop number j as a destination. A transport link corresponds to one possibility of the connection between two stops. Each edge has a weight t which is equal to the travel time (in minutes) between nodes i and j which can be determined on the base of timetables. A set of edges is bimodal because it includes, besides directed links, undirected walk links. The undirected edge $\{i, j, t\}$ is an element of the set W , if walk time in minutes between i and j stops is not greater than $limit_w$ parameter. The value of $limit_w$ parameter has a big influence on the number of network links (density of graph). The t value for undirected edge $\{i, j, t\}$ is equal to walk time in minutes between i and j stops.

A graph representation of public transportation network is shown in Fig. 1. It is a very simple example of the network which includes only nine stops. In the

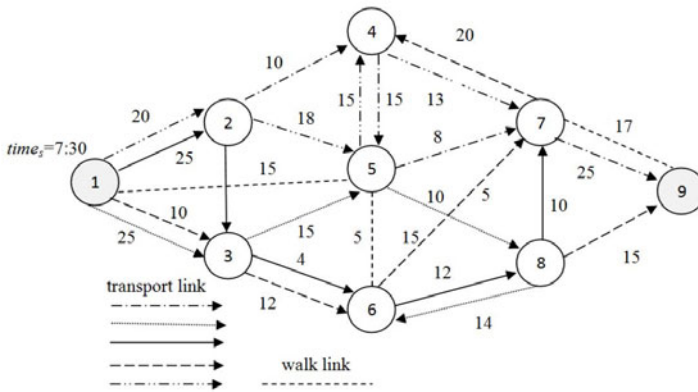


Fig. 1. Representation of a simple transportation network (different styles of lines mark different transport links; dot lines mark walk links; grey nodes mark the start (1) and the end stops (9))

real world the number of nodes is equal to 3500 for the city with about 1 million of inhabitants.

Formal definition of our problem is as follows: At the input we have: G - graph of transportation network, $timetable(l)$ - times of departures for each stops and line l , source point of the travel (o), destination point of the travel (d), starting time of the travel ($time_o$), number of the resultant paths (k) and limit for walk links ($limit_w$). At the output we want to have the set of resultant routes, containing at most k quasi-optimal paths with minimal time of realization (in minutes) in the first instance and with minimal number of transfers in the second.

Weight of transport link (i, j, l, t) is strongly dependent on the starting time parameter ($time_o$) and $timetable(l)$ which can be changed during the realization of the algorithm solving our problem. The t value of (i, j, l, t) link is equal to the result of subtraction: time of arrival for line l to the stop j and start time for stop i ($time_i$).

3 Description of EA

EA is a hybrid algorithm which combines a standard evolutionary technique [7] with application of results of pre-computations. Such combination takes effect in achieving of a final population with a high fitness for small size of initial population and small number of generations. Using EA routes determination is a multi-stage process involving following steps:

1. Pre-computation: Before first running of the EA two kind of sets of routes templates are determined: $T_{o,d}$ - sets of all possible templates of routes with the minimal number of transfers for each pair of o and d stops, $R_{o,d}$ - sets of all possible templates of routes with the minimal number of stops for each pair of o and d stops. A template of a route includes only consecutive stops, without travel time value on connections. We can determine these sets on the base of transfer graph G_{tr} . G_{tr} is the time independent graph $G_{tr} = \langle V, E \rangle$, where V is a set of stops and E is a set of directed transfer connections. The edge (i, j) is the transfer connection in G_{tr} , if there is at least one route without transfers between stops i and j . Methods for determining templates included in sets $T_{o,d}$ and $R_{o,d}$ are defined in [9]. We can transform very fast a given template to a route by determination travel time value for each connections of a route. An example of transformation templates to routes for the network presented in Fig. 1 is shown in Fig. 2. It's very important that time of realization of pre-computations doesn't increase the computation time of EA, because this step is using only one time before first running of the method.
2. Initialization: EA starts with generating an initial number of routes - P . Each route composes a chain of connections between considered consecutive stops. Two stops can be considered consecutive if they are consecutive stops for transport line or walk link. For each connection included in the route value of travel time t for a given start time $time(o)$ and timetables is determined. The

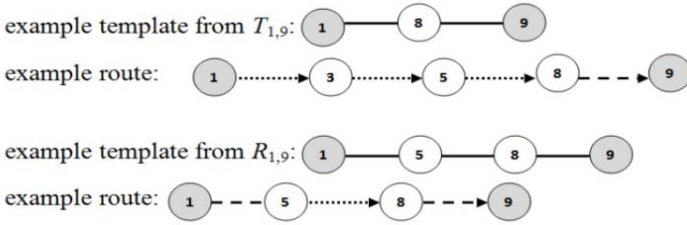


Fig. 2. (Example templates included in sets $T_{1,9}$ and $R_{1,9}$ and routes obtained from these templates for network shown in Fig. 1)

initial population is generated in a special way: m_1 individuals are computed as routes with minimal number of transfers, m_2 next individuals are routes with minimal number of stops and other $P - (m_1 + m_2)$ individuals are randomly generated routes without repeated stops. Individuals with minimal number of transfers are generated on the base of the set $T_{o,d}$. Chromosomes with minimal number of stops are generated on the based of the set $R_{o,d}$. If the size of set $T_{o,d}$ is less than m_1 or the size of set $R_{o,d}$ is less than m_2 then we must generate more random individuals.

3. Evaluation: We calculate a value of fitness function F to evaluate the optimum nature of each route (chromosome). The fitness function should estimate the quality of individuals, according to the time of realization of the tour and number of transfers in lexicographically order.
4. Improving population: After fitness evaluation, the EA starts to improve initial population through ng applications of crossover and mutation. In every generation we first choose with probability pr between crossover and mutation.

4.1. Crossover: We first select two parent individuals, according to the fitness value: the better an individual is, the bigger chance it has to be chosen. Since chromosomes lengths are different, we presented a new heuristic crossover operator, adjusted to our problem. In the first step we test if crossover can take place. If two parents do not have at least one common bus stop, crossover can not be done and parents remain unchanged. Crossover is implemented in the following way. First we choose one common bus stop, it will be the crossing point. Then we exchange fragments of tours from the crossing point to the end bus stop in two parent individuals. After crossover, we must correct offspring individuals in two ways. First we eliminate bus stop loops, then we eliminate line loops. The next step is to compute fitness function for these new individuals. Finally, we choose two best individuals from mutated chromosomes and offspring and add them to the population. Best individuals are individuals with the smallest travel time in first order and with minimal number of transfers in second order. The example of parents and offspring individuals after crossover operator is shown in Fig. 3.

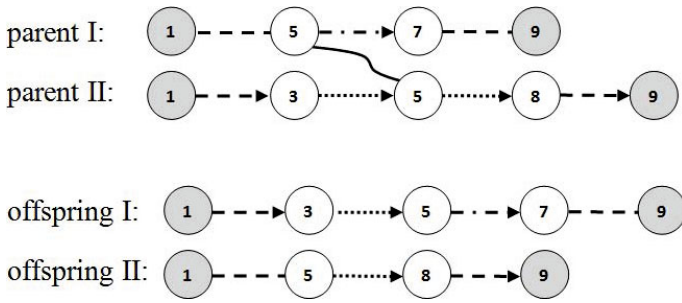


Fig. 3. (Example parents and offspring individuals after crossover for network shown in Fig. 1)

4.2. Mutation: We first choose randomly one chromosome. The next step is to randomly select two bus stops, denoted as o_1 and d_1 from the route ($o_1, d_1 \neq o, d$). Then we randomly choose k templates of routes from o_1 to d_1 with minimal number of transfers. From these templates we select a route with minimal time of realization. If there are more than one route with minimal travel time we select one with the smallest number of transfers. This best route exchanges the fragment of a route from o_1 to d_1 in a chromosome being mutated. Then we compute fitness function for this individual and add it to the population. The example of parent individuals and offsprings after mutation operator is shown in Fig. 4.

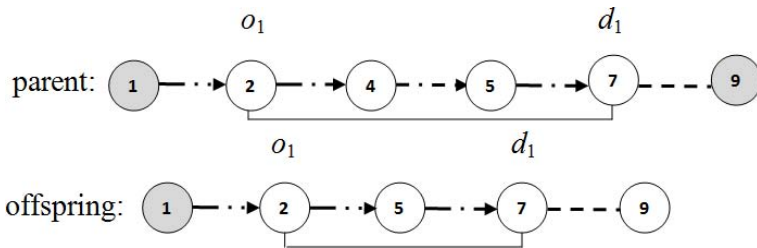


Fig. 4. (Example parent and offspring individual after mutation for network shown in Fig. 1)

5. Determining results: k resultant routes are selected from the final population by choosing k routes with the best fitness.

EA applies results of pre-computations in the initialization step and during the realization of the mutation operator. Therefore, initial routes and offsprings individuals after genetic operators have much better fitness than randomly generated chromosomes. Therefore, even for small size of population and number of generations its possible to determine good quality of resultant routes. Small values

of P and ng parameters have a big influence on the reduction of a computation time of the method.

4 Experimental Results

There were a number of computer tests conducted on real data of transportation network in Warsaw city. This network consists of about 4200 stops, connected by about 240 bus, tram and metro lines. Values of common parameters for EA, TG and CA algorithms were following: $k = 3$, $limit_w = 15$, $P \in \{10, 20, 30\}$, $ng \in \{10, 20, 30, 40, 50\}$. The value of $limit_w$ is very important because it influences the density of network. The bigger value of $limit_w$, the more possibilities of walk links in a network. Density of network is of a key importance for time-complexity of algorithms. Additional parameters only for EA were following: $pr = 0.5$, $m_1 = 30\%P$, $m_2 = 30\%P$. We examined routes from the center of the city to the periphery of the city (set CP), routes from the periphery of the city to the center of the city (set PC) and routes from the periphery of the city to the periphery of the city (set PP). Each of these sets includes 30 specification of first (o) and last (d) stops in the route which are difficult cases for each algorithm. First matter is a long distance from o and d (PP set), the second is a high density of the network in o or d localization (CP and PC sets). Algorithms was tested in a computer equipped with an Intel core2 Duo T7300, 2 GHz and 2 GB RAM.

In Tab. 1 are presented arithmetic averages of travel time of the best resultant route generated by EA, for considered sizes of population, numbers of generations and kinds of routes.

Table 1. Travel time of the best resultant routes generated by EA (for each value of ng), TG and CA

Routes- P	$ng = 10$	$ng = 20$	$ng = 30$	$ng = 40$	$ng = 50$	TG	CA
PC-10	55,98	53,98	50,67	50,67	49,98	65,98	53,84
CP-10	80,02	76,78	70,87	69,24	68,56	96,28	71,28
PP-10	96,78	94,23	90,24	88,14	88,14	106,55	93,33
PC-20	46,23	39,59	39,55	39,34	39,23	59,30	39,59
CP-20	72,76	61,00	60,05	59,67	59,56	74,15	62,33
PP-20	92,24	88,90	88,05	87,56	87,56	88,90	88,90
PC-30	45,89	39,45	39,35	38,67	38,67	44,32	36,55
CP-30	66,67	60,08	60,2	60,2	59,34	76,35	55,18
PP-30	91,89	87,24	87,12	86,25	86,25	99,11	89,14

In Tab. 2 are presented arithmetic averages of number of transfers of the best resultant route generated by EA, for considered sizes of population, numbers of generations and kinds of routes. We compared results obtained for EA with analogous results for CA and TG and it turn out that even than ng was equal 20 and P was equal 20 EA returns the best routes with a shorter travel time and

Table 2. Number of transfers of the best resultant routes generated by EA (for each value of ng), TG and CA

Routes- P	$ng = 10$	$ng = 20$	$ng = 30$	$ng = 40$	$ng = 50$	TG	CA
PC-10	4,08	3,67	3,55	3,55	3,46	4,77	3,64
CP-10	4,89	4,15	4,10	3,94	3,87	5,25	4,15
PP-10	5,78	5,50	5,30	5,27	5,27	6,53	5,45
PC-20	3,45	2,03	2,11	2,04	2,02	3,13	2,33
CP-20	3,56	2,63	2,63	2,66	2,66	2,63	2,63
PP-20	5,02	3,70	3,63	3,24	3,24	5,80	3,64
PC-30	2,45	2,01	2,15	2,15	2,15	4,15	2,01
CP-30	3,08	2,65	2,45	2,45	2,23	6,75	2,55
PP-30	4,67	3,71	3,67	3,56	3,56	5,86	3,60

smaller number of transfers than TG and comparable to CA. Moreover, for these values of parameters the computation time of EA was significantly shorter than for CA and TG. In Tab. 3 are shown average values of travel time (Avg-tt) and minimal number of transfers (Avg-tr) of the best route generated by EA, TG and CA for considered kind of routes. Results of EA are determined for $ng = 20$ and $P = 20$.

Table 3. The comparison of EA, TG and CA

Routes-method	Avg-tt	Avg-tr
PC-EA	39,59	2,03
CP-EA	61,00	2,63
PP-EA	88,90	3,70
PC-TG	57,11	2,63
CP-TG	75,60	6,70
PP-TG	115,40	6,81
PC-CA	38,58	1,48
CP-CA	61,45	2,2
PP-CA	90,80	2,33

The last experiment was focused on comparison of computation time of algorithms. The results are presented in Fig. 5. In this experiment we tested examples of routes with a minimal number of stops, between 5 and 61. On the horizontal axis there are points representing the minimal number of stops on a route. These values were computed as a result of standard *BFS* graph search method and they are correlated with difficulty of the route. On the vertical axis there is marked time of execution. Each possible route with a given number of the minimal number of stops was tested by three algorithms at starting time at 7:30 a.m., weekday. The computation time of algorithms was averaged over every tested routes. We can see in Fig. 5 that EA performs in significantly shorter time than CA and TG, especially for routes with minimal number of stops greater than 43.

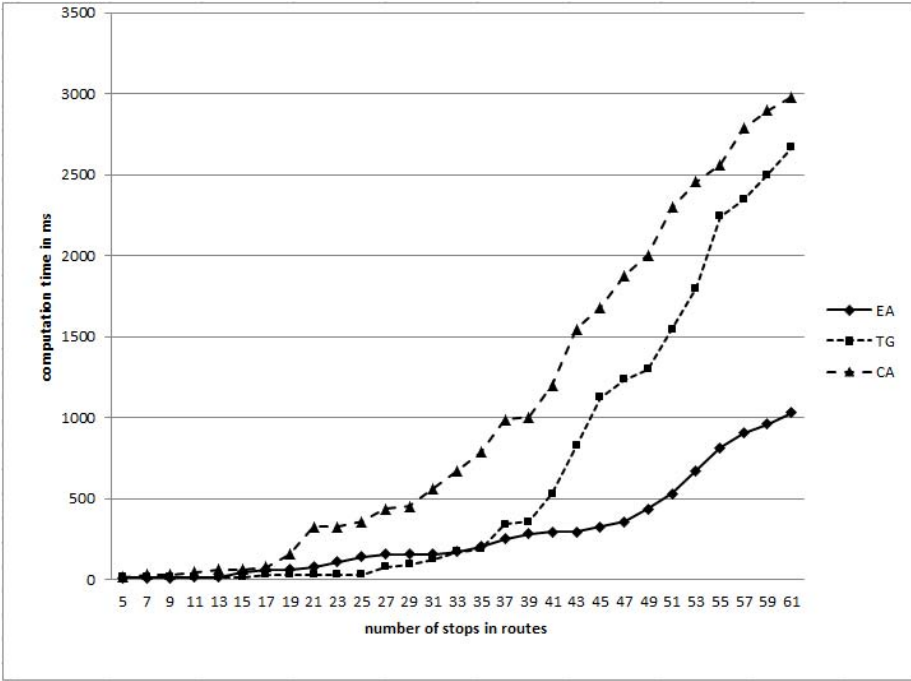


Fig. 5. The comparison of the execution time of EA, TG and CA

Both compared algorithms were tested only on small networks with number of bus-stops less than 1000 and therefore their computation time was satisfactory.

5 Conclusions

Computer experiments have shown that EA performs much better than comparable methods. As future work, it is intended to expand experimentation with other instances: big metropolises or regions with number of stops exceeding 5000 and small and/or rare networks. If tests show poor performance of EA the new heuristics must be added to the algorithm. The proposal of improvement which can be considered includes to the algorithm information about geographic location of start and destination stops.

Acknowledgments. This research was supported by S/WI/1/11.

References

1. Ahuja, R.K., Orlin, J.B., Pallotino, S., Scutella, M.G.: Dynamic shortest path minimizing travel times and costs. *Networks* 41(4), 197–205 (2003)

2. Bellman, R.E.: On a Routing Problem. *Journal Quarterly of Applied Mathematics* 16, 87–90 (1958)
3. Chabini, I.: Discrete dynamic shortest path problems in transportation applications. Complexity and algorithms with optimal run time, *Journal Transportation Research Records*, 170–175 (1998)
4. Cooke, K.L., Halsey, E.: The shortest route through a network with time-dependent intermodal transit times. *Journal Math. Anal. Appl.* 14, 493–498 (1998)
5. Dreyfus, S.E.: An Appraisal of Some Shortest-path Algorithms. *Journal Operations Research* 17, 395–412 (1969)
6. Galves-Fernandez C. Khadraoui D. and remainder: Distributed Approach for Solving Time-Dependent Problems in Multimodal Transport Networks. In: *Advanced in Operation Research*, (2009)
7. Goldberg, D.E.: Genetic algorithms and their applications. WNT, Warsaw (1995)
8. Hansen, P.: Bicriterion path problems. In: *Multicriteria Decision Making: Theory and Applications. Lecture Notes in Economics and Mathematical Systems*, vol. 177, pp. 236–245 (1980)
9. Koszelew, J.: Approximation method to route generation in public transportation network. *Polish Journal of Enviromental Studies* 17, 418–422 (2008)
10. Piwonska, A., Koszelew, J.: Evolutionary algorithms find routes in public transport network with optimal time of realization. In: Mikulski, J. (ed.) *TST 2010. CCIS*, vol. 104, pp. 194–201. Springer, Heidelberg (2010)
11. Lawler, E.L.: A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem. *Management Science* 18, 401–405 (1972)
12. Orda, A., Rom, R.: Shortest path and minimum - delay algorithms in networks with time-dependent edge-length. *Journal Assoc. Computer Mach.* 37(3), 607–625 (1990)
13. Reyes, L.C., Zezzatti, C.A.O.O., Santillán, C.G., Hernández, P.H., Fuerte, M.V.: A Cultural Algorithm for the Urban Public Transportation. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) *HAIS 2010. LNCS (LNAI)*, vol. 6077, pp. 135–142. Springer, Heidelberg (2010)
14. Zhang, Y., Shiyong, C., Jinfeng, L., Fu, D.: The Application of Genetic Algorithm in Vehicle Routing Problem. In: *Electronic Commerce and Security, International Symposium, International Symposium on Electronic Commerce and Security*, pp. 3–6 (2008)

Exploring Market Behaviors with Evolutionary Mixed-Games Learning Model

Yu Du¹, Yingsai Dong¹, Zengchang Qin^{1,2}, and Tao Wan²

¹ Intelligent Computing and Machine Learning Lab
School of Automation Science and Electrical Engineering
Beihang University, Beijing, China

² Robotics Institute, Carnegie Mellon University, USA
ydu1989@gmail.com, zcqin@andrew.cmu.edu

Abstract. The minority game (MG) is a simple model for understanding collective behavior of agents competing for a limited resource. In our previous work, we assumed that collective data can be generated from combination of behaviors of variant groups of agents and proposed the minority game data mining (MGDM) model. In this paper, to further explore collective behaviors, we propose a new behavior learning model called Evolutionary Mixed-games Learning (EMGL) model, based on evolutionary optimization of mixed-games, which assumes there are variant groups of agents playing majority games as well as the minority games. Genetic Algorithms then are used to optimize group parameters to approximate the decomposition of the original system and use them to predict the outcomes of the next round. In experimental studies, we apply the EMGL model to real-world time-series data analysis by testing on a few stocks from Chinese stock market and the USD-RMB exchange rate. The results suggest that the EMGL model can predict statistically better than the MGDM model for most of the cases and both models perform significantly better than a random guess.

1 Introduction

Agent-based experimental games have attracted much attention of scientists from different research areas to explore complex systems such as financial markets [1]. New research themes such as experimental economics [2], financial market modeling [3] and market mechanism designs [4] have been flourished in recent years. In financial market modeling, an economic market is regarded as a complex adaptive system (CAS), and people try to analyze the real market system of which agents with similar capability compete for limited resources. Every agent knows the history data of the market and decides how to trade based on global information. The Minority Game (MG) has been widely used to model the interactions among agents as a simplified version of a financial market [5].

In previous work [6], we assumed the existence of one “intelligent agent” who can take advantages of the game by learning from all other agents’ behaviors in minority games. In reality, it is always infeasible to obtain all records of agents’

choices in each round of the game. If we assume that the collective data are generated from the combination of variant groups of agents' behaviors - which is intuitively true, how can we decompose the collective data into the combinations of micro-level data is what hope to explore in the Minority Game Data Mining (MGDM) [7]. Genetic algorithms (GA) are used to optimize the agent group parameters to yield the best approximation of the original dynamic system. The MGDM model was applied to the real-world time-series data analysis by testing on its effectiveness in stock market predictions [7].

However, there are some weaknesses of using the MG model in real-world market data analysis. First, since all agents have the same memory length, the diversity of agents is limited. Second, in the real-world markets, some agents play the minority game, which are referred to as "foundation traders" who hope to maximize their profits; while others are just "trend chasers" who choose what the majority do (i.e. majority game). In order to establish an agent-based model which more closely approximate the real market, Gou [8,9] modifies the MG model by dividing agents into two groups: one group play the minority game and the other group play the majority game, thus this system is referred to as a 'mixed-game' model. Inspired by the 'mixed-game' model, we propose the Evolutionary Mixed-game Learning (EMGL) model. We divide the agents in the game into three diverse groups: (1) the agents who make random decisions; (2) agents who play minority games and (3) agents who play majority games. By applying genetic algorithms, we can model the behaviors of above three types of agents, thus analyze and estimate the resource-constrained environment parameters to maximize the approximation of the system outputs to the real-world test data. That is a new way to understand the relationship between micro-behaviors and macro-behaviors in complex dynamic systems.

This paper is structured as follows: Section 2 introduces the mixed-game model. In section 3, we propose the EMGL model that uses genetic algorithms to optimize the mixed-game model to discover the composition of agents and predict the macro-behaviors in the resource-constrained environment. In section 4, we apply the EMGL model to predict financial time-series in the stock market. We also compare the results of the EMGL model with the previous MGDM model and verify the effectiveness of this learning mechanism. Conclusions and discussions are given in the end.

2 Mixed-Games Model

The Minority Game (MG) [5] was originated from the El Farol Bar problem and formulated to analyze decision-making. In the MG, there are an odd number of players and each must choose one of two choices independently at each round of the game, winners are those on the minority side at last. There is no prior communication among players; the only information available is numbers of players corresponding to two choices of the last round. In this section, we will set a resource-constrained environment populated by three diverse types of agents: agents who make random decisions (random traders), agents who play minority

game, and agents who play majority game, representing so-called “trend chasers”. This variety of agents is for simulating a more realistic real market [8].

2.1 Strategies of Agents

Suppose an odd number of N agents decide between two possible options, say to attend Room A or B at each round of the game. Formally, at round $t(t = 1, 2, \dots, T)$: each agent takes an action $a_i(t)$ for $i = 1, 2, \dots, N$ to choose, i.e :

$$a_i(t) = \begin{cases} A & \text{Agent } i \text{ choose room } A \\ B & \text{Agent } i \text{ choose room } B \end{cases} \quad (1)$$

At each round t , agents belonging to the minority group win. The winning outcome can be represented by binary code function $w(t)$. If A is the minority side, i.e. the number of agents choosing Room A is no greater than $(N - 1)/2$, we define the winning outcome $w(t) = 0$; otherwise, $w(t) = 1$. In this paper, the winning outcomes are known to public, formally represented by:

$$w(t) = \begin{cases} 0 & \text{if: } \sum_{i=1}^N \Delta(a_i(t) = A) \leq (N - 1)/2 \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

where $\Delta(\alpha)$ is the truth function: if α is true, then $\Delta(\alpha)$ is 1; otherwise, $\Delta(\alpha)$ is 0. We assume that agents make choices based on the most recent m winning outcomes $h(t)$, which is called history memory and m is the memory length, formally: $h(t) = [w(t - m), \dots, w(t - 2), w(t - 1)]$.

In the MG, we usually assume that each agent’s reaction towards the previous data is governed by a “strategy” [5]. Each strategy is based on the past m -bit memory, described as a binary sequence, then there are 2^{2^m} possible strategies in the strategy space. Each agent looks into the most recent history for the same pattern of m bit string and predicts the outcome. Given history memory $h(t)$, we denoted Agent i ’s choice guided by strategy S as $S(h(t))$. Table 1 shows one possible strategy S with $m = 4$. For example, $h(t) = [0000]$ represents that if the winning outcomes of the latest 4 rounds are all 0, the next round (at round t) choice for this agent will be $S([0000]) = A$. Thus a strategy can be regarded as a particular set of decisions on the permutations of previous winning outcomes.

Table 1. One possible strategy S with the memory length $m = 4$

$h(t)$	0000	0001	0010	0011	0100	0101	0110	0111
$S(h(t))$	A	A	A	B	B	A	A	A
$h(t)$	1000	1001	1010	1011	1100	1101	1110	1111
$S(h(t))$	A	B	B	B	B	A	A	B

2.2 Mixed-Games with Heterogeneous Agents

In order to obtain a better approximation of the collective behaviors in the real-world market, Gou [8,9] modifies the MG model and proposes the ‘mixed-game model’, in which agents are divided into two groups: each group has different memory length, Group G_N plays minority game with the same strategy, while Group G_J plays majority game with the same strategy. Comparing to the MG, the most significant part of mixed-game is that it has an additional group of “trend chasers”, therefore be more realistic to simulate a real-world market.

Given a training time range, all agents in G_N choose the best strategy with which they can predict the minority side most correctly, while all agents in G_J choose the best strategy with which they can predict the majority side most correctly. N_1 represents the number of agents in G_N and N_2 represents the number of agents in G_J . We use m_1 and m_2 , respectively, to describe the memory length of these two groups of agents. As each agent’s reaction is based on a strategy corresponding a response to past memories, there are $2^{2^{(m_1)}}$ and $2^{2^{(m_2)}}$ possible strategies for G_N or G_J , respectively.

However, the mixed-game model is still not a reasonable prediction tool in real-world market with the following reasons: in real-life scenarios, it is unrealistic to assume all agents playing minority game or majority game hold the same strategy and follow the same rule: some agents make random decisions and different subgroups hold different strategies; if all agents act in the same way, they will all lose. We assume the completeness of marketing world is embodied in existence of variant groups of agents using their own strategies. Therefore, in this paper, we improve the mixed-game by dividing the agents into three diverse types of agents: agents who make random decisions (denoted by G_R), agents of Group G_N (playing the minority game) with different strategies, agents of Group G_J (playing the majority game) with different strategies.

3 Evolutionary Mixed-Games Learning Model

As we mentioned above, we propose a framework based on the assumption that the macro-behavior of the market is an aggregation of three groups of agents:

- Group G_N : Agents who play minority game.
- Group G_J : Agents who play majority game.
- Group G_R : Agents who make random decisions.

For G_N and G_J we assume that the overall effect can be decomposed into several small subgroups, while each subgroup of agents use a certain strategy. The decomposition of the collective behaviors involves a big set of parameters including the number of agents in each subgroup and the strategies they employ. We aim to use genetic algorithms to tune these parameters for these subgroups of agents to yield the collective behavior has the best approximation of the history data.

3.1 Chromosome Encoding

In our model, we use a parameter vector to represent the number of agents of each subgroup and the corresponding strategy they use, then we apply GA to explore the most likely combinations of subgroup behaviors that could generate the best approximated macro-level sequences. Given the history winning outcomes $w(t)$, the expected maximum number of subgroups using fixed strategies in G_N is K_N , and the expected maximum number of subgroups using fixed strategies in G_J is K_J . Thus agents of the whole system can be divided into $K_N + K_J + 1$ groups:

$$\{G_R, G(S_N^1), G(S_N^2), \dots, G(S_N^{K_N}), G(S_J^1), G(S_J^2), \dots, G(S_J^{K_J})\}$$

where G_R represents the group of random agents, $G(S_N^i)$ (for $i = 1, \dots, K_N$) represents the subgroup agents holding strategy S_N^i in Group G_N (the group playing minority game). $G(S_J^k)$ (for $k = 1, \dots, K_J$) represents the subgroup agents holding strategy S_J^k in Group G_J .

The chromosome for genetic algorithms \mathbf{x} is encoded with the following parameters: $\mathbf{x} = \{P_R, P(S_N^1), S_N^1, \dots, P(S_N^{K_N}), S_N^{K_N}, P(S_J^1), S_J^1, \dots, P(S_J^{K_J}), S_J^{K_J}\}$

- P_R : the percentage of random agents among all agents (i.e. $P_R = \frac{G_R}{N}$)
- $P(S_N^i)$: the percentage of the number of agents in the minority game subgroup i ($i \in [1, 2, \dots, K_N]$) with the fixed strategy S_N^i (i.e. $P(S_N^i) = \frac{|G(S_N^i)|}{N}$).
- S_N^i : Binary coding of the minority game strategy S_N^i .
- $P(S_J^k)$: the percentage of the number of agents in the majority game subgroup k ($k \in [1, 2, \dots, K_J]$) with the fixed strategy S_J^k (i.e. $P(S_J^k) = \frac{|G(S_J^k)|}{N}$).
- S_J^k : Binary coding of the majority game strategy S_J^k .

Figure 1 illustrates that the collective behavior is a combination of choices from the above three types of agents. Given history sequence $h(t)$, the intelligent agent can use GA to explore all possible combinations of subgroups and compositions of the market, then use the information to make choice on the minority side.

3.2 Fitness Function

In the EMGL model, we aim to generate a system in which agents from both Group G_N and Group G_J can achieve their goals to the greatest extent, i.e., agents in Group G_N end up on the minority side while agents in Group G_J end up on the majority side. The final goal of EMGL model aims to obtain the best prediction of the market and make rational choice to maximum its profits. At round t , in order to evaluate the chromosome \mathbf{x}_j ($j = 1, 2, \dots, J$ where J is the population size), we run the mixed-game with parameter setting decoded from \mathbf{x}_j and get the prediction outcome. We choose the best chromosome by calculating the fitness function $f(\mathbf{x}_j)$ with the following three rules:

At round t , we consider collective data within the previous T steps: $(t - 1 - T, t - T, \dots, t - 2, t - 1)$

- *Rule 1*: For all agents in Group G_N , every time an agent predicts the correct outcome, i.e. chooses on the minority side, we add one point to $f(\mathbf{x}_j)$.

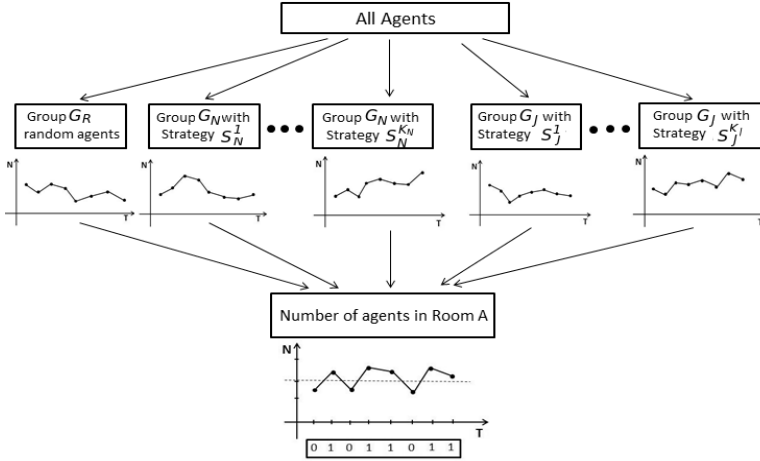


Fig. 1. The process of generating collective data. All agents can be divided into $K_N + K_J + 1$ groups where agents in the same subgroups act the same based on the strategy they follow. The collective data can be regarded as an aggregation of all agents’ actions.

- *Rule 2:* For all agents in Group G_J , every time an agent predicts the correct outcome, i.e. chooses on the majority side, we add one point to $f(\mathbf{x}_j)$.
- *Rule 3:* If the prediction outcome $y_i(t)$ by the EMGL model is equal to the real-world macro outcome $w(t)$, we add a specific weight $W_{predict}$ to $f(\mathbf{x}_j)$.

Usually we set the weight value as a specific percentage of the total number of agents N : $W_{predict} = \beta N$ ($\beta \in [0, 1]$).

We calculate the fitness function $f(\mathbf{x}_j)$ for $t_0 = t - 1 - T, t - T, \dots, t - 2, t - 1$ and select the best chromosome \mathbf{x}_j^* within the time range T .

$$\mathbf{x}^*(t) = \arg \max_j f(\mathbf{x}_j(t)) \quad for \quad j = 1, \dots, J \tag{3}$$

Then we decode parameters from the best chromosome to obtain the best prediction of whole system and choose to be on the minority side.

4 Experiments on Real-World Markets

The EMGL model points a new way of using mixed-games model and evolutionary optimization in understanding the relationship between micro-data and macro-data. Given a sequence of history winning outcomes, we can use GA to explore the most likely combinations of single behaviors that could generate this sequence. Many real-world complex phenomena are caused by aggregations of agents’ behaviors such as stock market and currency exchange rate, which are regarded as random and unpredictable in classical economics. In the following experiments, we apply the EMGL model to explore the compositions of the system

Table 2. Comparisons of mean prediction accuracy of the EMGL and MGDM [7] models on 12 real-world financial time-series data including 11 stocks from Chinese market and the USD-RMB exchange rate

# Data	Stock index	Start from (m/d/y)	MGDM in [7]	EMGL (4-3)	EMGL (5-3)	EMGL (6-3)
1. USD-RMB Exchange Rate	-	Jan 02 2001	58.59%	63.78%	64.13%	62.51%
2. SPD Bank Co.	600000	Jan 02 2001	55.99%	52.41%	50.17%	51.63%
3. Shandong Bohui Paper Co.	600966	Jun 08 2004	53.94%	54.45%	56.22%	54.54%
4. Shenergy Co.	600642	Jan 02 2001	51.78%	49.70%	49.61%	49.87%
5. China Minsheng Banking Co.	600016	Dec 19 2000	55.71%	52.63%	54.66%	52.44%
6. Qingdao Haier Co.	600690	Jan 02 2001	49.52%	54.01%	54.56%	54.02%
7. Huaneng Power Industrial Inc.	600011	Dec 06 2000	50.87%	51.23%	51.62%	51.21%
8. China United Network Comm.	600050	Oct 09 2002	51.34%	54.38%	53.83%	54.59%
9. CNTIC Trading Co.	600056	May 15 1997	52.99%	54.84%	55.09%	54.53%
10. Hisense Electric Co.	600060	Apr 22 1997	53.13%	56.93%	56.79%	57.68%
11. China Television Media Ltd	600088	Jun 16 1997	50.69%	52.11%	54.14%	53.29%
12. China Eastern Airlines Co.	600115	Nov 05 1997	55.62%	57.14%	56.69%	56.44%

using agents playing mixed-games. We can tune the parameters by training on the history data and use these estimated parameters to make future predictions.

4.1 Experiment Design

In the following experiments, We randomly select 11 stocks from the Chinese stock market through a downloadable software¹, and also the U.S.Dollar-RMB (Chinese Renminbi) exchange rate². Compared with the validation method used in our previous work [7], we use a different validation benchmark in this paper: for each stock or currency exchange rate, we use the winning outcomes from 1 – 500 trading days as training set to obtain relatively adaptable chromosomes, and then predict financial time-series of 501 – 800 trading days. We compare the result of EMGL with MGDM to test the effectiveness of the new model.

Each round of the game represents one trading day. Given macro-level data $w(t)$, the best chromosome $\mathbf{x}^*(t)$ is selected and the parameter information in \mathbf{x}^* is used for predicting the winning choice in the next round. Suppose the opening price is V_b and the closing price is V_f . For each trading day t , fluctuation of the stock price or exchange rate can be transferred to $w(t)$ as follows: if $V_b > V_f$, then $w(t) = 1$; otherwise, $w(t) = 0$. By correctly predicting $w(t)$ using the learning model, we can capture the ups and downs of the market prices.

In the following experiments with the MGDM and EMGL models, we set $K_N = K_J = 20$. Since almost all agents play with history memories of 6 or less in a typical MG [10], and m_N is usually larger than m_J when using mixed-game model to simulate real market [8], we set $m_N = 4, 5, 6$ and $m_J = 3$ to

¹ Website: http://big5.newone.com.cn/download/new_zszq.exe

² Data obtained from: <http://bbs.jjxj.org/thread-69632-1-7.html>

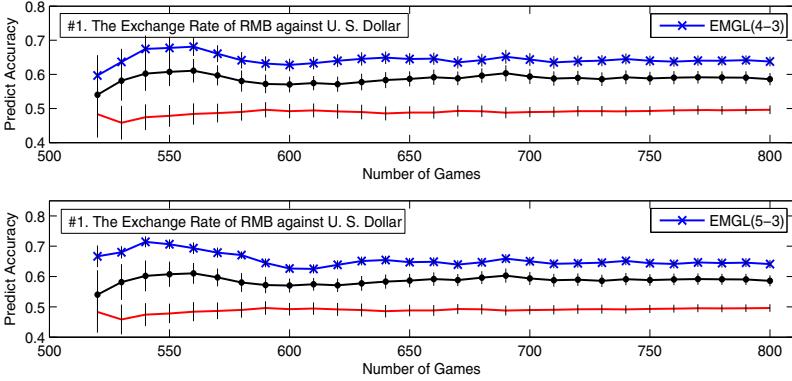


Fig. 2. Performance of the MGDM model and the EMGL model with different memory lengths on the USD-RMB exchange rate

establish three configuration of EMGL models. We set $K = 20$ and $m = 3$ in the MGDM model. As for the GA, we set population size $J = 50$, crossover rate $P_c = 0.8$, mutation rate $P_m = 0.05$, the specific weight $\beta = 0.5$. We run the whole experiments for 30 times to reduce the influences of randomness in GAs.

4.2 Data Analysis

Table 2 shows the prediction accuracy of 11 stocks and the US Dollar-RMB exchange rate within 501 – 800 trading days, where EMGL(6-3) represents $m_N = 6, m_J = 3$, etc. We use different configurations of memory length ($m_N = 4, 5, 6; m_J = 3$) and calculate the mean prediction accuracy and its standard deviations. For most of the cases, the EMGL model performs statistically better than the MGDM model (for 8 of 11 stocks and U.S.Dollar-RMB exchange rate). By adding agents who play majority game, we can generate a more realistic market and predict the stock prices more accurately.

From the USD-RMB experiment shown in Figure 2, we can see both EMGL (starred curve) and MGDM (dotted curve) can predict with high accuracy (the mean accuracy is up to 58.6% for MGDM and 63.8% for EMGL (4-3)), indicating a strong existing pattern captured by the new models. In general, almost all results of MGDM and EMGL are statistically better than the random guess (the mean is around 50% with a small variance) plotted at the bottom.

Figure 3 shows the performance on stock # 10, 12 and 5, which are three representational results in the experiments. Like the experimental results on USD-RMB exchange rate, the test on stock # 10 shows that the EMGL model outperform the GMDM model and both models outperform the random guess. 7 of 12 data (# 1, 3, 6, 8, 9, 10, 11) have similar performance. In the experiments on stock # 12 (and 7), the EMGL model and the MGDM model have similar accuracy (which are not statistically different from each other) and both models

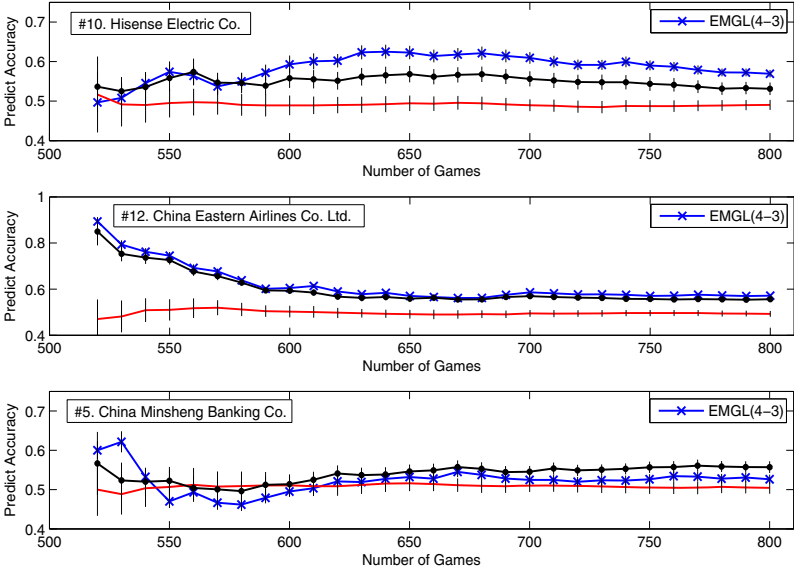


Fig. 3. Performance of the MGDM model and the EMGL model on three representational stocks # 10, 12 and 5

outperform the random guess. For stock # 5 (and 4), two prediction curves are overlapped, therefore we are not able to tell which model is statistically better. For stock # 2, the MGDM model outperform the EMGL model which means that the minority game modeling could be more appropriate than mixed-games in this case. The stock prices are driven by complex behaviors and influenced by many unknown factors, it is hard to tell what sort of micro-behavior could be more appropriate than others. However, empirical results on these data have shown that the proposed learning framework of collective data decomposition is effective in solving this difficult problem. Though the EMGL model performs statistically better than the MGDM model for most of the cases in our experiments, we still need to be cautious about choosing between the MGDM model and the EMGL model (as well as different configurations of memory lengths), the performance of these two models may vary with specific stocks when making predictions of the market. The computation time of 30 rounds of GAs on the given 12 dataset is about 5 hours using the Matlab code on an Intel Pentium dual-core PC.

5 Conclusions

In this paper, we proposed a novel learning framework of considering the collective market behavior is an aggregation of several subgroup of agents' behaviors based on the mixed-games model. By using GAs to explore all the possibilities of decomposition of the system, the new model is capable in predicting time-series

data and make decisions to maximize its profits. We tested the EMGL model on a few real-world stock data and the USD-RMB exchange rate. For most of the cases, the EMGL model performs statistically better than the MGDm model and both models perform significantly better than a random guess. The future work will focus on obtaining the real returns on more stocks in market. We are also interested in analyzing the correlations between different memory length configurations of the EMGL model.

Acknowledgment. This work is partially funded by the NCET Program of MOE, China, the SRF for ROCS and the China Scholar Council.

References

1. Mantegna, R., Stanley, H.: *An Introduction to Econophysics: Correlations and Complexity in Finance*. Cambridge University Press, Cambridge (1999)
2. Gode, D., Sunder, S.: Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy* 101(1), 119–137 (1993)
3. Johnson, N., Jefferies, P., Hui, P.: *Financial Market Complexity*. Oxford University Press, Oxford (2003)
4. Qin, Z.: Market mechanism designs with heterogeneous trading agents. In: *Proceedings of Fifth International Conference on Machine Learning and Applications (ICMLA)*, Orlando, Florida, USA, pp. 69–74 (2006)
5. Challet, D., Zhang, Y.: Emergence of cooperation in an evolutionary game. *Physica A: Statistical and Theoretical Physics* 246(3-4), 407–418 (1997)
6. Li, G., Ma, Y., Dong, Y., Qin, Z.: Behavior learning in minority games. In: Guttman, C., Dignum, F., Georgeff, M. (eds.) *CARE 2009/2010*. LNCS (LNAI), vol. 6066, pp. 125–136. Springer, Heidelberg (2011)
7. Ma, Y., Li, G., Dong, Y., Qin, Z.: Minority Game Data Mining for Stock Market Predictions. In: Cao, L., Bazzan, A.L.C., Gorodetsky, V., Mitkas, P.A., Weiss, G., Yu, P.S. (eds.) *ADMI 2010*. LNCS, vol. 5980, pp. 178–189. Springer, Heidelberg (2010)
8. Gou, C.: *Dynamic Behaviors of Mix-game Models and Its Application* (2005), <http://arxiv.org/abs/physics/0504001>
9. Gou, C.: Agents Play Mix-game. In: *Econophysics of Stock and Other Markets*. LNCS, Part II, pp. 123–132 (2006)
10. Savit, R., Koelle, K., Treynor, W., Gonzalez, R.: In: Tumer, Wolpert (eds.) *Collectives and the Design of Complex System*, pp. 199–212. Springer, Heidelberg (2004)

On the Web Ontology Rule Language OWL 2 RL

Son Thanh Cao¹, Linh Anh Nguyen², and Andrzej Szalas^{2,3}

¹ Faculty of Information Technology, Vinh University
182 Le Duan street, Vinh, Nghe An, Vietnam
sonct@vinhuni.edu.vn

² Institute of Informatics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
{nguyen,andsz}@mimuw.edu.pl

³ Dept. of Computer and Information Science, Linköping University
SE-581 83 Linköping, Sweden

Abstract. It is known that the OWL 2 RL Web Ontology Language Profile has PTIME data complexity and can be translated into Datalog. However, a knowledge base in OWL 2 RL may be unsatisfiable. The reason is that, when translated into Datalog, the result may consist of a Datalog program and a set of constraints in the form of negative clauses. In this paper we first identify a maximal fragment of OWL 2 RL called OWL 2 RL⁺ with the property that every knowledge base expressed in this fragment can be translated into a Datalog program and hence is satisfiable. We then propose some extensions of OWL 2 RL and OWL 2 RL⁺ that still have PTIME data complexity.

1 Introduction

Semantic Web is a rapidly growing research area that has received lots of attention from researchers in the last decade. One of the layers of Semantic Web is OWL (Web Ontology Language), which is used to specify knowledge of the domain in terms of concepts, roles and individuals. The second version OWL 2 of OWL, recommended by the W3C consortium in October 2009, is based on the description logic *SR_QIQ* [7]. This logic is highly expressive but has intractable combined complexity (N2EXPTIME-complete) and data complexity (NP-hard) for basic reasoning problems. Thus, W3C recommended also profiles of OWL 2 with lower expressiveness and tractable complexity.

OWL 2 RL [13] is one of such profiles. It restricts the full language OWL 2 to allow a translation into Datalog and hence to obtain PTIME data complexity and efficient computational methods. However, a knowledge base in OWL 2 RL may be unsatisfiable. Namely, the Datalog program resulting from the translation may also contain a set of constraints expressed as negative clauses. In database applications, constraints are important for keeping the database consistent w.r.t. its specification. They are used to block data modifications that are not allowed. In Semantic Web applications, the situation is a bit different. An ontology may consist of many component ontologies which were developed independently by

different people. It may also include component ontologies which will be updated independently by different people. Thus, constraints are less important for ontologies, and when possible, should be replaced by syntactic restrictions. This is the case of OWL 2 RL.

In this paper we first identify a maximal fragment of OWL 2 RL called OWL 2 RL⁺ with the property that every knowledge base expressed in this fragment can be translated into a Datalog program and hence is satisfiable. We then propose an extension OWL 2 eRL of OWL 2 RL with PTIME data complexity, and an extension OWL 2 eRL⁺ of OWL 2 RL⁺ that can be translated into eDatalog (an extended version of Datalog). Next, we extend both OWL 2 eRL and OWL 2 eRL⁺ with eDatalog. Combining OWL 2 eRL or OWL 2 eRL⁺ with eDatalog gives us the freedom to use the syntax of both the languages and allows us to represent knowledge about the domain not only in terms of concepts and roles but also by using predicates with higher arities.

Related Work: OWL 2 RL has been inspired by Description Logic Programs (DLP) [5] and pD^* [14]. The logical base of DLP is the description Horn logic DHL [5]. Some extensions of DHL were studied by Nguyen in [11]. There are a number of fragments of description logics with PTIME data complexity (see [12, Section 4] for an overview). Some other related works are [6,4] (on description logic programs with negation), [2] (on layered rule-based architecture) and [9,10,3] (on Horn fragments of modal logics).

The Structure of this Paper: In Section 2 we specify OWL 2 RL [13] as a logical formalism. Section 3 is devoted to OWL 2 RL⁺. Section 4 presents extensions of OWL 2 RL and OWL 2 RL⁺. Section 5 concludes this work. Due to the lack of space, proofs of our theorems are presented only in [1].

2 The Logical Formalism of OWL 2 RL

In this section we specify OWL 2 RL as a logical formalism, using the syntax of description logics. We strictly follow the specification of OWL 2 RL given in [13].

We denote the set of *concept names* by CNames, and the set of *role names* by RNames. We use the truth symbol \top to denote *owl:Thing* [13], and use:

- a and b to denote *individuals* (i.e. *objects*)
- d to denote a *literal* [13] (i.e. a data constant)
- A and B to denote concept names (i.e. *Class* elements [13])
- C and D to denote *concepts* (i.e. *ClassExpression* elements [13])
- lC to denote a concept standing for a *subClassExpression* of [13]
- rC to denote a concept standing for a *superClassExpression* of [13]
- eC to denote a concept standing for an *equivClassExpression* of [13]
- DT to denote a *data type* (i.e. a *Datatype* of [13])
- DR to denote a *data range* (i.e. a *DataRange* of [13])
- r and s to denote *object role names* (i.e. *ObjectProperty* elements [13])
- R and S to denote *object roles* (i.e. *ObjectPropertyExpression* elements [13])
- σ and ρ to denote *data role names* (i.e. *DataProperty* elements [13]).

The families of R , DR , lC , rC , eC are defined by the following BNF grammar:

$$\begin{aligned}
R &:= r \mid r^- \\
DR &:= DT \mid DT \sqcap DR \\
lC &:= A \mid \{a\} \mid lC \sqcap lC \mid lC \sqcup lC \mid \exists R.lC \mid \exists R.\top \mid \exists \sigma.DR \mid \exists \sigma.\{d\} \\
rC &:= A \mid rC \sqcap rC \mid \neg lC \mid \forall R.rC \mid \exists R.\{a\} \mid \forall \sigma.DR \mid \exists \sigma.\{d\} \mid \\
&\quad \leq 1 R.lC \mid \leq 0 R.lC \mid \leq 1 R.\top \mid \leq 0 R.\top \mid \leq 1 \sigma.DR \mid \leq 0 \sigma.DR \\
eC &:= A \mid eC \sqcap eC \mid \exists R.\{a\} \mid \exists \sigma.\{d\}
\end{aligned}$$

The class constructor *ObjectOneOf* [13] can be written as $\{a_1, \dots, a_k\}$ and expressed as $\{a_1\} \sqcup \dots \sqcup \{a_k\}$.

We will use the following abbreviations: Disj (Disjoint), Func (Functional), InvFunc (InverseFunctional), Reff (Reflexive), Irref (Irreflexive), Sym (Symmetric), Asym (Asymmetric), Trans (Transitive), Key (HasKey).

A *TBox axiom*, which stands for a *ClassAxiom* or a *DatatypeDefinition* or a *HasKey* axiom [13], is an expression of one of the following forms:

$$\begin{aligned}
lC \sqsubseteq rC, \quad eC = eC', \quad \text{Disj}(lC_1, \dots, lC_k), \quad DT = DR, \\
\text{Key}(lC, R_1, \dots, R_k, \sigma_1, \dots, \sigma_h).
\end{aligned}$$

An *RBox axiom*, which stands for an *ObjectPropertyAxiom* or a *DataPropertyAxiom* [13], is an expression of one of the following forms:

$$\begin{aligned}
R_1 \circ \dots \circ R_k \sqsubseteq S, \quad R = S, \quad R = S^-, \quad \text{Disj}(R_1, \dots, R_k), \quad \exists R.\top \sqsubseteq rC, \quad \top \sqsubseteq \forall R.rC, \\
\text{Func}(R), \quad \text{InvFunc}(R), \quad \text{Irref}(R), \quad \text{Sym}(R), \quad \text{Asym}(R), \quad \text{Trans}(R), \\
\sigma \sqsubseteq \varrho, \quad \sigma = \varrho, \quad \text{Disj}(\sigma_1, \dots, \sigma_k), \quad \exists \sigma \sqsubseteq rC, \quad \top \sqsubseteq \forall \sigma.DR, \quad \text{Func}(\sigma).
\end{aligned}$$

Note that axioms of the form $R = S$, $R = S^-$, $\text{Sym}(R)$ or $\text{Trans}(R)$ are expressible by axioms of the form $R_1 \circ \dots \circ R_k \sqsubseteq S$, and hence can be deleted from the above list. An RBox axiom of the form $\exists R.\top \sqsubseteq rC$ (resp. $\top \sqsubseteq \forall R.rC$, $\exists \sigma \sqsubseteq rC$, $\top \sqsubseteq \forall \sigma.DR$) stands for an *ObjectPropertyDomain* (resp. *ObjectPropertyRange*, *DataPropertyDomain*, *DataPropertyRange*) axiom [13]. One can classify these latter axioms as TBox axioms instead of RBox axioms. Similarly, $\text{Key}(\dots)$ axioms can be classified as RBox axioms instead.

An *ABox assertion* is a formula of one of the following forms:

$$a = b, \quad a \neq b, \quad rC(a), \quad DT(d), \quad R(a, b), \quad \neg R(a, b), \quad \sigma(a, d), \quad \neg \sigma(a, d).$$

In OWL 2 RL [13], assertions of the form $DT(d)$ are implicitly provided by declarations of DT and d . The other ABox assertions stand for *Assertion* elements of [13]. We also call an ABox assertion as an *ABox axiom*.

In OWL 2 RL [13], there are also axioms standing for declarations and annotation axioms and used for keeping meta information about the ontology. These kinds of axioms are inessential from the logical point of view and hence are omitted here.

An *RBox* (resp. *TBox*, *ABox*) is a finite set of RBox (resp. TBox, ABox) axioms. An ABox \mathcal{A} is said to be *extensionally reduced* if it does not contain

$$\begin{aligned}
\{d\}^{\mathcal{I}} &= \{d^{\mathcal{I}}\}, & (DT \sqcap DR)^{\mathcal{I}} &= DT^{\mathcal{I}} \cap DR^{\mathcal{I}} \\
(R^{-})^{\mathcal{I}} &= (R^{\mathcal{I}})^{-1} = \{(y, x) \mid (x, y) \in R^{\mathcal{I}}\} \\
\top^{\mathcal{I}} &= \Delta_o^{\mathcal{I}}, & \{a\}^{\mathcal{I}} &= \{a^{\mathcal{I}}\}, & (\neg C)^{\mathcal{I}} &= \Delta_o^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}}, & (C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} &= \{x \in \Delta_o^{\mathcal{I}} \mid \forall y[(x, y) \in R^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}]\} \\
(\exists R.C)^{\mathcal{I}} &= \{x \in \Delta_o^{\mathcal{I}} \mid \exists y[(x, y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}]\} \\
(\forall \sigma.DR)^{\mathcal{I}} &= \{x \in \Delta_o^{\mathcal{I}} \mid \forall y[(x, y) \in \sigma^{\mathcal{I}} \text{ implies } y \in DR^{\mathcal{I}}]\} \\
(\exists \sigma.\varphi)^{\mathcal{I}} &= \{x \in \Delta_o^{\mathcal{I}} \mid \exists y[(x, y) \in \sigma^{\mathcal{I}} \text{ and } y \in \varphi^{\mathcal{I}}]\} \\
(\exists \sigma)^{\mathcal{I}} &= \{x \in \Delta_o^{\mathcal{I}} \mid \exists y(x, y) \in \sigma^{\mathcal{I}}\} \\
(\leq n R.C)^{\mathcal{I}} &= \{x \in \Delta_o^{\mathcal{I}} \mid \#\{y \in \Delta_o^{\mathcal{I}} \mid (x, y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \leq n\} \\
(\leq n \sigma.DR)^{\mathcal{I}} &= \{x \in \Delta_o^{\mathcal{I}} \mid \#\{y \in \Delta_d^{\mathcal{I}} \mid (x, y) \in \sigma^{\mathcal{I}} \text{ and } y \in DR^{\mathcal{I}}\} \leq n\}
\end{aligned}$$

Fig. 1. Interpretation of data ranges, inverse object roles, and complex concepts. In this figure, φ is of the form DR or $\{d\}$, and $\#\Gamma$ denotes the cardinality of the set Γ .

axioms of the form $C(a)$ with C being a complex concept (i.e., not a concept name). A *knowledge base* (i.e. an ontology) in OWL 2 RL is defined to be a tuple $(\mathcal{R}, \mathcal{T}, \mathcal{A})$ consisting of an RBox \mathcal{R} , a TBox \mathcal{T} , and an ABox \mathcal{A} . We may present a knowledge base as a set of axioms.

An *interpretation* $\mathcal{I} = \langle \Delta_o^{\mathcal{I}}, \Delta_d^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ consists of a non-empty set $\Delta_o^{\mathcal{I}}$ called the *object domain* of \mathcal{I} , a non-empty set $\Delta_d^{\mathcal{I}}$ disjoint with $\Delta_o^{\mathcal{I}}$ called the *data domain* of \mathcal{I} , and a function $\cdot^{\mathcal{I}}$ called the *interpretation function* of \mathcal{I} , which maps

- every individual a to an element $a^{\mathcal{I}} \in \Delta_o^{\mathcal{I}}$
- every literal d to an element $d^{\mathcal{I}} \in \Delta_d^{\mathcal{I}}$
- every concept name A to a subset $A^{\mathcal{I}}$ of $\Delta_o^{\mathcal{I}}$
- every data type DT to a subset $DT^{\mathcal{I}}$ of $\Delta_d^{\mathcal{I}}$
- every object role name r to a binary relation $r^{\mathcal{I}} \subseteq \Delta_o^{\mathcal{I}} \times \Delta_o^{\mathcal{I}}$
- every data role name σ to a binary relation $\sigma^{\mathcal{I}} \subseteq \Delta_o^{\mathcal{I}} \times \Delta_d^{\mathcal{I}}$.

As OWL 2 RL has no features for declaring whether two literals are equal or not, we adopt the unique name assumption for literals, which means that if $d_1 \neq d_2$ then $d_1^{\mathcal{I}} \neq d_2^{\mathcal{I}}$. The interpretation function is extended to interpret data ranges, inverse object roles and complex concepts as shown in Figure 1.

From now on, if not stated otherwise, by an *axiom* we mean an RBox axiom, a TBox axiom or an ABox axiom. The satisfaction relation $\mathcal{I} \models \varphi$ between an interpretation \mathcal{I} and an axiom φ is defined below and stands for “ \mathcal{I} validates φ ”:

- $\mathcal{I} \models R_1 \circ \dots \circ R_k \sqsubseteq S$ iff $R_1^{\mathcal{I}} \circ \dots \circ R_k^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
- $\mathcal{I} \models C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- $\mathcal{I} \models C(a)$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$
- $\mathcal{I} \models R(a, b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

- $\mathcal{I} \models \sigma(a, d)$ iff $(a^{\mathcal{I}}, d^{\mathcal{I}}) \in \sigma^{\mathcal{I}}$
- $\mathcal{I} \models \varphi = \psi$ iff $\varphi^{\mathcal{I}} = \psi^{\mathcal{I}}$,
where φ and ψ may be of the form C, R, R^-, DT, DR or a
- $\mathcal{I} \models a \neq b$ iff $a^{\mathcal{I}} \neq b^{\mathcal{I}}$
- $\mathcal{I} \models \text{Disj}(\varphi_1, \dots, \varphi_k)$, where $\varphi_1, \dots, \varphi_k$ are of the form C, R or σ ,
iff, for all $1 \leq i < j \leq k$, $\varphi_i^{\mathcal{I}} \cap \varphi_j^{\mathcal{I}} = \emptyset$
- $\mathcal{I} \models \text{Func}(R)$ iff $R^{\mathcal{I}}$ is functional (i.e. $\forall x, y, z (R^{\mathcal{I}}(x, y) \wedge R^{\mathcal{I}}(x, z) \rightarrow y = z)$)
- $\mathcal{I} \models \text{InvFunc}(R)$ iff $R^{\mathcal{I}}$ is inverse-functional
(i.e. $\forall x, y, z (R^{\mathcal{I}}(x, z) \wedge R^{\mathcal{I}}(y, z) \rightarrow x = y)$)
- $\mathcal{I} \models \text{Irref}(R)$ iff $R^{\mathcal{I}}$ is irreflexive
- $\mathcal{I} \models \text{Sym}(R)$ iff $R^{\mathcal{I}}$ is symmetric
- $\mathcal{I} \models \text{Asym}(R)$ iff $R^{\mathcal{I}}$ is asymmetric
- $\mathcal{I} \models \text{Trans}(R)$ iff $R^{\mathcal{I}}$ is transitive
- $\mathcal{I} \models \text{Func}(\sigma)$ iff $\sigma^{\mathcal{I}}$ is functional
- $\mathcal{I} \models \text{Key}(C, R_1, \dots, R_k, \sigma_1, \dots, \sigma_h)$ iff, for every $x, y \in C^{\mathcal{I}}, z_1, \dots, z_k \in \Delta_o^{\mathcal{I}}$
and $d_1, \dots, d_h \in \Delta_d^{\mathcal{I}}$, if $\{(x, z_i), (y, z_i)\} \subseteq R_i^{\mathcal{I}}$ and $\{(x, d_j), (y, d_j)\} \subseteq \sigma_j^{\mathcal{I}}$ for
all $1 \leq i \leq k$ and $1 \leq j \leq h$, then $x = y$.

The semantics of $\text{Key}(C, R_1, \dots, R_k, \sigma_1, \dots, \sigma_h)$ is defined according to the semantics of the *HasKey* constructor of [13], but note that it has a clear meaning only when all the roles $R_1, \dots, R_k, \sigma_1, \dots, \sigma_h$ are assumed to be functional.

When φ is an ABox axiom, we also say \mathcal{I} *satisfies* φ to mean \mathcal{I} validates φ .

An interpretation \mathcal{I} is called a *model* of an RBox, a TBox or an ABox if it validates all the axioms of the box. \mathcal{I} is called a model of a knowledge base $(\mathcal{R}, \mathcal{T}, \mathcal{A})$, denoted by $\mathcal{I} \models (\mathcal{R}, \mathcal{T}, \mathcal{A})$, if it is a model of all \mathcal{R}, \mathcal{T} and \mathcal{A} .

A (*conjunctive*) *query* is a formula of the form $\varphi_1 \wedge \dots \wedge \varphi_k$, where each φ_i is an ABox assertion. An interpretation \mathcal{I} satisfies a query $\varphi = \varphi_1 \wedge \dots \wedge \varphi_k$, denoted by $\mathcal{I} \models \varphi$, if $\mathcal{I} \models \varphi_i$ for all $1 \leq i \leq k$. We say that a query φ is a *logical consequence* of a knowledge base $(\mathcal{R}, \mathcal{T}, \mathcal{A})$, denoted by $(\mathcal{R}, \mathcal{T}, \mathcal{A}) \models \varphi$, if every model of $(\mathcal{R}, \mathcal{T}, \mathcal{A})$ satisfies φ .

Note that, queries are defined to be “ground”. In a more general context, queries may contain variables for individuals or literals. However, one of the approaches to deal with such queries is to instantiate variables by individuals or literals occurring in the knowledge base or the query.

The *data complexity* of OWL 2 RL (for the conjunctive query answering problem) is the complexity of checking where a query φ is a logical consequence of a knowledge base $(\mathcal{R}, \mathcal{T}, \mathcal{A})$, measured w.r.t. the size of the ABox \mathcal{A} when assuming that \mathcal{A} is extensionally reduced and \mathcal{R}, \mathcal{T} and φ are fixed.

In [1] we present examples of unsatisfiable knowledge bases in OWL 2 RL.

3 The Fragment OWL 2 RL⁺

We define OWL 2 RL⁺ to be the restriction of OWL 2 RL such that:

- the constructors $\neg lC, \leq 0 R.lC, \leq 0 R.\top$ and $\leq n \sigma.DR$ (where n is 0 or 1) are disallowed in the BNF grammar rule defining the family of rC

- axioms of the forms $\text{Disj}(\dots)$, $\text{Irref}(R)$, $\text{Asym}(R)$, $a \neq b$, $\neg R(a, b)$, $\neg\sigma(a, d)$ are disallowed.

These restrictions correspond to the following ones for the OWL 2 RL of [13]:

- the constructors *superComplementOf*, *superObjectMaxCardinality* with limit 0, and *superDataMaxCardinality* are disallowed in the definition of *superClassProperty*
- axioms of the following kinds are disallowed
 - *DisjointClasses*, *DisjointObjectProperties*, *DisjointDataProperties*,
 - *IrreflexiveObjectProperty*, *AsymmetricObjectProperty*,
 - *DifferentIndividuals*,
 - *NegativeObjectPropertyAssertion*, *NegativeDataPropertyAssertion*.

A query is said to be *in the language of KB* if it does not use predicates not occurring in *KB*. A *positive query* is a formula of the form $\varphi_1 \wedge \dots \wedge \varphi_k$, where each φ_i is an ABox assertion of one of the forms $a = b$, $rC(a)$, $R(a, b)$, $\sigma(a, d)$.

We now recall definitions of Datalog:

- A *term* is either a *constant* or a *variable*.
- If p is an n -array predicate and t_1, \dots, t_n are terms then $p(t_1, \dots, t_n)$ is an *atomic formula*, which is also called an *atom*.
- A *Datalog program clause* is a formula of the form $\varphi_1 \wedge \dots \wedge \varphi_n \rightarrow \psi$, where $n \geq 0$ and $\varphi_1, \dots, \varphi_n, \psi$ are atoms. The conjunction $\varphi_1 \wedge \dots \wedge \varphi_n$ is called the *body* and ψ is called the *head* of the clause. The program clause is required to satisfy the *range-restrictedness* condition, which states that every variable occurring in the clause’s head must occur also in the clause’s body.
- A *Datalog program* is a finite set of Datalog program clauses.

Theorem 3.1

1. *OWL 2 RL⁺ is a maximal fragment (w.r.t. allowed features) of OWL 2 RL such that every knowledge base expressed in the fragment is satisfiable.*
2. *Every knowledge base KB in OWL 2 RL⁺ can be translated into a Datalog program P which is equivalent to KB in the sense that, for every query φ in the language of KB, KB ⊨ φ iff P ⊨ φ.* ◁

Let \perp stands for the falsity symbol, with the semantics that $\perp^I = \emptyset \subset \Delta_o^I$.

Let *KB* be a knowledge base in OWL 2 RL. The *normal form of KB* is the knowledge base obtained from *KB* as follows: if $\neg lC$ occurs as an rC in the knowledge base then replace it by a fresh (new) concept name A and add to the knowledge base the axiom $A \sqcap lC \sqsubseteq \perp$. The *corresponding version of KB in OWL 2 RL⁺* is the knowledge base obtained from the normal form of *KB* by deleting all axioms of the forms $A \sqcap lC \sqsubseteq \perp$, $\text{Disj}(\dots)$, $\text{Irref}(R)$, $\text{Asym}(R)$, $a \neq b$, $\neg R(a, b)$, $\neg\sigma(a, d)$.

Theorem 3.2. *Let KB be a knowledge base in OWL 2 RL, KB’ be the normal form of KB, and KB’’ be the corresponding version of KB in OWL 2 RL⁺. Then:*

1. KB' is equivalent to KB in the sense that, for every query φ in the language of KB , $KB \models \varphi$ iff $KB' \models \varphi$
2. if KB is satisfiable and φ is a positive query in the language of KB then $KB \models \varphi$ iff $KB' \models \varphi$. ◁

The second assertion states that if KB is satisfiable then the corresponding version of KB in OWL 2 RL⁺ is equivalent to KB w.r.t. positive queries. This means that, ignoring constraints and considering only positive queries, OWL 2 RL can be replaced by OWL 2 RL⁺ without any further loss of generality.

4 Extensions of OWL 2 RL with PTIME Data Complexity

In this section we first define an extension of Datalog called eDatalog. We then propose an extension OWL 2 eRL of OWL 2 RL with PTIME data complexity, and an extension OWL 2 eRL⁺ of OWL 2 RL⁺ that can be translated into eDatalog. Next, we extend both OWL 2 eRL and OWL 2 eRL⁺ with eDatalog.

4.1 eDatalog

From the point of view of OWL, there are two basic types: *individual* (i.e. *object*) and *literal* (i.e. *data*). For simplicity, we do not divide the *literal* type into smaller ones like natural numbers, real numbers, strings. We denote the *individual* type by *IType*, and the *literal* type by *LType*. Thus, a concept name is a unary predicate of type $P(\text{IType})$, an object role name is a binary predicate of type $P(\text{IType} \times \text{IType})$, and a data role name is a binary predicate of type $P(\text{IType} \times \text{LType})$. Extending OWL 2 RL with Datalog, apart from concept names and role names we use also a set OPreds of *ordinary predicates* and a set ECPreds of *external checkable predicates*. We assume that the sets CNames, RNames, OPreds and ECPreds are pairwise disjoint. A k -ary predicate from OPreds has type $P(T_1 \times \dots \times T_k)$, where each T_i is either *IType* or *LType*. A k -ary predicate from ECPreds has type $P(\text{LType}^k)$. We assume that each predicate from ECPreds has a fixed meaning which is checkable in the sense that, if p is a k -ary predicate from ECPreds and d_1, \dots, d_k are constant elements of *LType*, then the truth value of $p(d_1, \dots, d_k)$ is fixed and computable. For example, one may want to use the binary predicates $>$, \geq , $<$, \leq on real numbers with the usual semantics. We assume there are two different equality predicates, both denoted by '='. The first one belongs to OPreds and has the type $P(\text{IType} \times \text{IType})$. The second one is a binary predicate belonging to ECPreds with the usual semantics according to the unique name assumption.

Extending Datalog to eDatalog, we want to drop the range-restrictedness condition. However, to allow external checkable predicates we cannot do so totally. For this reason, we distinguish a subset RRPreds \subseteq CNames \cup RNames \cup OPreds as the set of *range-restricted predicates*. We define eDatalog as follows:

- A *term* is either an individual (of type *IType*) or a literal (of type *LType*) or a *variable* (of type *IType* or *LType*). If p is a predicate of type $P(T_1 \times \dots \times T_k)$,

- and for $1 \leq i \leq k$, t_i is a term of type T_i , then $p(t_1, \dots, t_k)$ is an *atomic formula* (also called an *atom*). An atom is *ground* if it contains no variables.
- An *eDatalog program clause* is a formula of the form $\varphi_1 \wedge \dots \wedge \varphi_n \rightarrow \psi$, where $n \geq 0$ and $\varphi_1, \dots, \varphi_n, \psi$ are atomic formulas such that:
 - ψ is an atom of a predicate from $\text{CNames} \cup \text{RNames} \cup \text{OPreds}$
 - if the predicate of ψ belongs to RRPreds then every variable occurring in ψ occurs also in some φ_i whose predicate also belongs to RRPreds
 - every variable occurring in some φ_i whose predicate belongs to ECPreds occurs also in some atom φ_j whose predicate belongs to RRPreds
 - An *eDatalog program* is a finite set of eDatalog program clauses.
 - A *knowledge base in eDatalog* is a pair $(\mathcal{P}, \mathcal{A})$, where \mathcal{P} is an eDatalog program and \mathcal{A} is an *ABox* consisting of ground atoms of predicates from $\text{CNames} \cup \text{RNames} \cup \text{OPreds}$.

Related notions for eDatalog like interpretation, model and data complexity are defined in the usual way, assuming the usual semantics for ‘=’ and the unique name assumption for literals (i.e. data constants).

4.2 OWL 2 eRL and OWL 2 eRL⁺

Axioms of the form $\text{Refl}(R)$ (i.e. reflexive object property axioms) are disallowed for OWL 2 RL. The reason is probably that translating $\text{Refl}(R)$ into Datalog we get a program clause $\forall x R(x, x)$ which violates the range-restrictedness condition. Similarly, \top is disallowed as *IC* in OWL 2 RL. However, these restrictions are unnecessary. The Horn fragment without function symbols of predicate logic also has PTIME data complexity. Furthermore, as shown in [8], evaluation methods of Datalog can be extended to Horn knowledge bases without function symbols in predicate logic. Thus, our first proposal is to extend OWL 2 RL with the feature of *ReflexiveObjectProperty* axioms and allowing \top as *IC*.

Our second proposal for extending OWL 2 RL is to allow unary predicates from ECPreds to appear in the places of *DataRange* elements. For example, it is desirable to express concepts like the class of all laptops with price not greater than 1000 USD. Using the syntax of description logic, the concept can be written as $\text{laptop} \sqcap \exists \text{price} . (\leq 1000)$. Here, “ ≤ 1000 ” is a unary predicate. Other useful predicates are the other comparison operators, the *between* operator, the operator used for checking pattern of a string. RDF or XML syntax for these operators and for constructing complex checkable predicates should be standardized.

By *OWL 2 eRL* we denote the extension of OWL 2 RL according to the two above mentioned proposals. By *OWL 2 eRL⁺* we denote the extension of OWL 2 RL⁺ by allowing axioms of the form $\text{Refl}(R)$ (i.e. *ReflexiveObjectProperty* axioms), allowing \top as *IC*, and allowing unary predicates from ECPreds to appear in the places of *DR* in the BNF grammar rule defining *IC*. Clearly, OWL 2 eRL⁺ is a sublanguage of OWL 2 eRL.

The data complexity of OWL 2 eRL or OWL 2 eRL⁺ is defined as usual.

Theorem 4.1

1. The languages *OWL 2 eRL* and *OWL 2 eRL⁺* have PTIME data complexity.
2. Every knowledge base KB in *OWL 2 eRL⁺* can be translated into a knowledge base KB' in *eDatalog* which is equivalent to KB in the sense that, for every query φ in the language of KB , $KB \models \varphi$ iff $KB' \models \varphi$. \triangleleft

4.3 Combining OWL 2 eRL and OWL 2 eRL⁺ with eDatalog

For the combined languages *OWL 2 eRL-eDatalog* and *OWL 2 eRL⁺-eDatalog* studied here we assume that all data role names belong to RRPreds (i.e. are range-restricted predicates). A *knowledge base* in the combined language *OWL 2 eRL-eDatalog* (resp. *OWL 2 eRL⁺-eDatalog*) is a tuple $(\mathcal{R}, \mathcal{T}, \mathcal{P}, \mathcal{A})$, where \mathcal{R} is an RBox of *OWL 2 eRL* (resp. *OWL 2 eRL⁺*), \mathcal{T} is a TBox of *OWL 2 eRL* (resp. *OWL 2 eRL⁺*), \mathcal{P} is an *eDatalog* program, and \mathcal{A} is a set consisting of ABox assertions of *OWL 2 eRL* (resp. *OWL 2 eRL⁺*) and ground atoms of ordinary predicates (from OPreds). The set \mathcal{A} is called an *ABox* and its elements are called *ABox assertions*.

A (*conjunctive*) *query* to a knowledge base of *OWL 2 eRL-eDatalog* (resp. *OWL 2 eRL⁺-eDatalog*) is a formula of the form $\varphi_1 \wedge \dots \wedge \varphi_k$, where each φ_i is an ABox assertion of *OWL 2 eRL-eDatalog* (resp. *OWL 2 eRL⁺-eDatalog*).

Other related notions are defined in the usual way.

Theorem 4.2

1. The combined languages *OWL 2 eRL-eDatalog* and *OWL 2 eRL⁺-eDatalog* have PTIME data complexity.
2. Given a knowledge base $KB = (\mathcal{R}, \mathcal{T}, \mathcal{P}, \mathcal{A})$ in *OWL 2 eRL⁺-eDatalog*, the set $\mathcal{R} \cup \mathcal{T}$ can be translated into an *eDatalog* program \mathcal{P}' such that KB is equivalent to the *eDatalog* knowledge base $(\mathcal{P}' \cup \mathcal{P}, \mathcal{A})$ in the sense that, for every query φ in the language of KB , $KB \models \varphi$ iff $\mathcal{P}' \cup \mathcal{P} \cup \mathcal{A} \models \varphi$. \triangleleft

Example 4.3. This example involves car insurance discounts. It comes from [11]. Consider the knowledge base in *OWL 2 eRL⁺-eDatalog* with $\mathcal{R} = \emptyset$ and

$$\begin{aligned} \mathcal{T} &= \{ \exists \text{has_child}.\top \sqsubseteq \text{parent}, \\ &\quad \text{parent} \sqcap \text{male} \sqsubseteq \text{father}, \\ &\quad \text{parent} \sqcap \text{female} \sqsubseteq \text{mother} \} \\ \mathcal{P} &= \{ \text{father}(x) \wedge \text{has_child}(x, y) \wedge \text{age}(y, k) \wedge k \leq 3 \rightarrow \text{discount}(x, 10), \\ &\quad \text{mother}(x) \wedge \text{has_child}(x, y) \wedge \text{age}(y, k) \wedge k \leq 3 \rightarrow \text{discount}(x, 15) \} \\ \mathcal{A} &= \{ \text{female}(\text{Jane}), \text{male}(\text{Mike}), \text{male}(\text{Peter}), \\ &\quad \text{has_child}(\text{Jane}, \text{Peter}), \text{has_child}(\text{Mike}, \text{Peter}), \text{age}(\text{Peter}, 2) \}. \end{aligned}$$

The query $\text{discount}(x, y)$ to this knowledge base has answers $(\text{Jane}, 15)$ and $(\text{Mike}, 10)$. (The discounts are in percentage.) \triangleleft

5 Conclusions

We have identified the maximal fragment OWL 2 RL⁺ of OWL 2 RL with the property that every knowledge base expressed in this fragment is satisfiable. We have also proposed extensions of OWL 2 RL and OWL 2 RL⁺ with PTIME data complexity by allowing *ReflexiveObjectProperty* axioms, external checkable predicates, eDatalog program clauses, and allowing \top as *IC*. These extensions are novel and very natural. They allow efficient computational methods (based on the ones of Datalog) and are useful for practical applications of Semantic Web.

Acknowledgements. This work was supported by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the strategic scientific research and experimental development program: “Interdisciplinary System for Interactive Scientific and Scientific-Technical Information”.

References

1. Cao, S.T., Nguyen, L.A., Szalas, A.: The long version of this paper (2011), Available at <http://www.mimuw.edu.pl/~nguyen/owl2r1-long.pdf>
2. Dunin-Kępicz, B., Nguyen, L.A., Szalas, A.: A layered rule-based architecture for approximate knowledge fusion. *Computer Science and Information Systems* 7(3), 617–642 (2010)
3. Dunin-Kępicz, B., Nguyen, L.A., Szalas, A.: Tractable approximate knowledge fusion using the Horn fragment of serial propositional dynamic logic. *Int. J. Approx. Reasoning* 51(3) (2010)
4. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R.: Well-founded semantics for description logic programs in the Semantic Web. *ACM Trans. Comput. Log.* 12(2), 11 (2011)
5. Grosz, B.N., Horrocks, I., Volz, R., Decker, S.: Description logic programs: combining logic programs with description logic. In: *Proceedings of WWW 2003*, pp. 48–57 (2003)
6. Heymans, S., Eiter, T., Xiao, G.: Tractable reasoning with DL-programs over Datalog-rewritable description logics. In: Coelho, H., Studer, R., Wooldridge, M. (eds.) *Proceedings of ECAI 2010*, pp. 35–40. IOS Press, Amsterdam (2010)
7. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible SROIQ. In: Doherty, P., Mylopoulos, J., Welty, C.A. (eds.) *Proceedings of KR 2006*, pp. 57–67. AAAI Press, Menlo Park (2006)
8. Madalińska-Bugaj, E., Nguyen, L.A.: Generalizing the QSQR evaluation method for Horn knowledge bases. In: Nguyen, N.T., Katarzyniak, R. (eds.) *New Challenges in Applied Intelligence Technologies. Studies in Computational Intelligence*, vol. 134, pp. 145–154. Springer, Heidelberg (2008)
9. Nguyen, L.A.: On the deterministic Horn fragment of test-free PDL. In: Hodkinson, I., Venema, Y. (eds.) *Advances in Modal Logic*, pp. 373–392. King’s College Publications (2006)
10. Nguyen, L.A.: Constructing finite least Kripke models for positive logic programs in serial regular grammar logics. *Logic Journal of the IGPL* 16(2), 175–193 (2008)

11. Nguyen, L.A.: Extending the description Horn logic DHL. In: Czaja, L., Szczuka, M. (eds.) Proceedings of CS&P 2009, pp. 419–430 (2009)
12. Nguyen, L.A.: Horn knowledge bases in regular description logics with PTime data complexity. *Fundamenta Informaticae* 104(4), 349–384 (2010)
13. OWL 2 RL (2009), http://www.w3.org/TR/owl2-profiles/#OWL_2_RL
14. ter Horst, H.J.: Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. *J. Web Sem.* 3(2-3), 79–115 (2005)

Results of Research on Method for Intelligent Composing Thematic Maps in the Field of Web GIS

Piotr Grobelny and Andrzej Pieczyński

University of Zielona Gora, Faculty of Electrical Engineering,
Computer Science and Telecommunications,
Podgorna 50, 65-246 Zielona Gora, Poland
P.Grobelny@weit.uz.zgora.pl, A.Pieczynski@issi.uz.zgora.pl

Abstract. Many of today's Web GIS applications are mashups, as they involve contents and functions from multiple Web services. This article presents a study on the use of expert system for discovery and matchmaking the Internet complex services in the field of Geographic Information Systems. To perform the experiment, a computer system has been prepared, to accomplish the method proposed by the author. Thereafter research results were presented to indicate how useful the method was and the effectiveness of the expert system use in solving the raised issues.

Keywords: Web GIS, Semantic Web Services, Expert System, Mashup Applications, OGC Standards.

1 Introduction

Unimaginable amount of information available on the Internet has led researchers to work on the next generation network known as the Semantic Web [2], in which published data is given meaning by attaching semantic description of its content. Thus processing contextual information by computers becomes possible. This "understanding" of information involves the use of ontology to define new concepts.

The next natural step was to transfer these experiences to the Web services [7]. Using semantics to describe the behavior and the capacity of Internet services it has arisen technology called the Semantic Web Services, allowing you to perform tasks such as discovery, matchmaking and execution of services, supplied by different vendors scattered throughout the global network.

The implementation of the tasks mentioned above leads to the development of decision support systems. These systems enhanced with artificial intelligence methods combine the capabilities of collecting and processing large amounts of data, the use of more diverse models and intelligent use of the accumulated knowledge. Thanks to this it is possible to analyze the data and automatically draw conclusions in a manner close to the human way of thinking – also using uncertain or fuzzy data. The main objective of building an intelligent system is an adequate representation of the existing knowledge, needed to solve the problem.

A very important use case is taking different decisions based on spatial analysis. Geographic Information Systems (GIS) provide access to data by creating maps on

demand. Depending on your needs, you can select interesting objects and present them in a form of thematic maps [11]. Linking descriptive and spatial information allows the system to support intelligent decision-making "belonging" to the human experts in such areas as crisis management, planning, urban and regional development, searching for specific locations (POI).

Given the recent rapid development of Internet GIS systems (Web GIS), authors of this paper decided to combine Semantic Web Services approach with the chosen technique of artificial intelligence (expert system) to develop adaptive methods of map building [9],[10] as mashup applications [8], [14]. It is assumed that, based on user's queries in a form close to the natural language, the computer system will generate a map "on demand". The role of the expert system will be to find and propose Web services which provide the required elements of map.

1.1 Problem

Grids and application servers allowed the next stage of the global network development in a form of remote sharing software in Software as a Service model (SaaS). The growing number of Web services maintained in enterprise networks, but also widely available on the Internet, has created a need to retrieve them on the basis of their functional and non-functional properties (semantic description of service) and composing them into the new complex functionalities.

In order to solve this generally introduced problem the authors of this paper have proposed their own method of matchmaking comprehensive Internet-based services using an expert system in GIS area. Similar methods are described in the literature with regard to Web services, but representing business processes [6], [16]. New rapidly developing areas of information systems, are Web GIS, which allow managing spatial information via a Web browser. The difference is that within the business processes these Web services are elements of sequenced chain of functionality, with strictly defined preconditions and effects. In the field of GIS sequence of each service does not matter. They are all parts of specific thematic maps, presented simultaneously to the user. The map processing is more similar to the processing of hypertext documents, where a description of the resource concerns mainly the content presented by it. It may be said that the approach presented by the authors is at the meeting point of the Semantic Web and Semantic Web Services technologies.

This requires identification of more specific problems: A) Proposing more appropriate service specification model for the Semantic Web GIS, according to which a knowledge base of the expert system will be created; B) The creation of Domain Specific Language (DSL) specific for the selected domain, allowing the submission of queries to the rules engine in order to run an expertise presenting the compound service. Due to the nature of the GIS services previously described (human-machine communication) this language should have a form similar to the natural language, C) Identification of the type and class of expert systems, in which it is possible to create a knowledge base on the basis of semantic service descriptions and submission of queries by using language designed by the authors. The system should have adequate quality, expressed as the main criterion for such a high level of expertise, to be issued in the time acceptable to the user.

1.2 Related Work

Semantic specification of services is fundamental to create the knowledge base of an expert system. Many researchers [7], [12], [17] indicate the rules as an important paradigm for representing knowledge about the Semantic Web Services. Using decision rules the knowledge can be expressed as: *IF a THEN b*, using an ontology, which allows the presentation of knowledge in the context of the given domain by using concepts, their attributes and relations.

In literature the foundation for the development of Semantic Web Services technology [7], [13] can be found. These approaches are focused on the composition of comprehensive services within the business processes, but geographical services are arranged in a different way. Complex service in this case contains many items that are displayed simultaneously in a Web browser. Information which links them all together is the geographic position of the presented artifacts. Therefore, for this new category of information systems, the authors offered own model of semantic description of a Web service, which complements this field of knowledge.

The obvious direction of Semantic Web Services development was to automate Web services composition. The authors of [3], [7], [17] define declarative programming based on Prolog language as the paradigm for drawing conclusions on the basis of knowledge about the semantic Web services. The F-Logic [12] language can be an example proposed as an ontology language based on Prolog rules for the Semantic Web. Another approach presented by authors of this paper which has not been applied in the considered area of knowledge concerns so called production systems. These are constructed on the basis of production rules which reflect the primary objective of human reasoning and implementation of activities in the light of stimuli [15]. According to this principle, production rule takes a form of an expression: *WHEN conditions THEN actions*. In this formula conditions determine the rule implementation, and the conclusions determine the triggered action. Production rules are very useful in cases where knowledge is composed of loosely related facts and independent actions. The process of handling production rules is called pattern matching [15]. This allows to record facts (semantic specification of services) in the knowledge base in a form of decision rules and inference using the production rules expressed in the DSL.

Domain Specific Language is a programming language tailored to the specific needs of computer applications. DSL used in the described method is a way of creating queries to the expert system in the context of the given area of knowledge. It draws on principles of programming based on production rules [15]. They allow you to interact with the system through transparent expressions, which are similar to the form of natural language. Achievement of the authors of this publication is to provide unique grammar rules for the language specific for the domain which allows inquiring the deductive database of expert system, using elements of the natural language closely related to the terminology of the field of GIS.

2 System Architecture

Elements of the system (based on SOA) used to conduct the experiment, and links between GIS, Web services and expert system (a registry of services) are shown in Fig. 1.

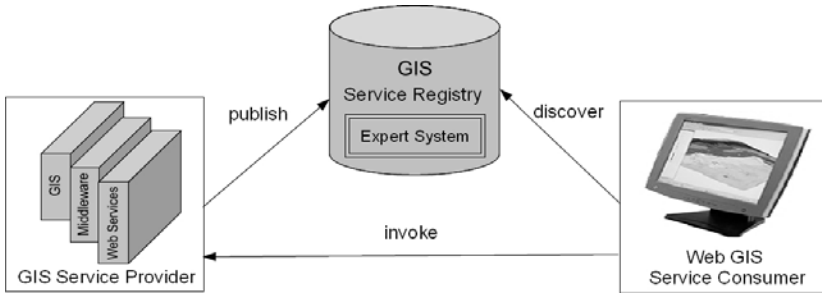


Fig. 1. System architecture based on SOA

2.1 Service Provider

Service provider has a server on which Web services are installed and controls access to them. It is also responsible for publishing a description of the service in the registry. Information systems which use digital maps (GIS) provide equivalents of traditional paper maps on a computer screen, but much more functional. Their distinguishing feature is the separation of storage place of the digital map (vector or raster) with selected objects, from the alphanumeric data describing these objects. Both of these data types are available to the service provider using software called geo-server made in technology known as middleware and available to the services recipients (consumers) through standardized WS-* interfaces.

2.2 Service Consumer

The service consumer is an application acting on behalf of the user. It allows the user to perform the task by locating and running (invoking) the relevant services and matchmaking them as a GIS mashup application [8], [14]. The end user requires a convenient graphical user interface, which is a form of overlay on a very extensive functionality of the GIS system. Its purpose is to present the maps in a form of image files or stored in vectors (SVG). Currently, the directions of development of such interfaces are applications which run in the Web browser environment. They allow presentation and selection of the data in any ration (zoom in, zoom out, choice of layers and areas, etc.). The most important requirement for the service recipient is the use of interoperable standards accepted by the broad community in the field of GIS (OGC standards) [8], [14]. This ensures correct calling and "consuming" of the Web services provided by the service provider.

2.3 Service Registry

In the described system architecture (Fig. 1) the expert system acts as a service registry. This is a special repository which allows publication of semantic specifications by the service provider on one hand, and on the other, finding and using the service by the consumer. It represents a kind of deductive database, which allows you to search the atomic Web services based on records of their semantic descriptions

(knowledge base). Its queries allow searching a set of services, all from different providers, which meet customer's requirements and to display them together on the computer screen.

3 Method

The objective of creating GIS systems is to gather information on a topic related geographically. Selective access to these systems' data allows creating a map on demand. It is a map generated from a database, which represents only selected thematic layers. Depending on your needs, you can select interesting objects and present them together in a form of thematic maps. Many users can use the same GIS database to create maps tailored especially to their requirements [11].

Described method is used to create such thematic maps. Expert system works as an advisor, which on the basis of user queries, proposes appropriate components of the map (thematic layers). These components are supplied by servers distributed on the Internet as Web services. Many layers of information representing various artifacts can be displayed simultaneously in the Web browser. It can be said that the proposed method allows the adaptive map building. This means that the system user can create a different map showing different information each time, by sending different queries e.g. *show the areas distant from the city of Zielona Gora by 10 km* or *show the road between the city of Zielona Gora, a the city of Berlin*.

Elements of the proposed method can be divided into two groups. The first group are elements which establish the system of knowledge representation such as: domain ontology, semantic service description model, knowledge base. The second group includes elements related to the inference mechanisms i.e. a strategy and algorithm of reasoning and domain specific language (DSL). Full development of the above issues and description of ways of combining the method elements have been carried out by authors in a previous article [10]. This document focuses on the description of a computer simulation to verify the assumptions and to present the results of the experiment. For the purposes of experimental verification of the method an integrated expert system has been designed and implemented. Inference process is realized by the rules engine and consists of determining conclusions and results on a basis of the data – facts in a form of decision rules (1) – stored in the knowledge base

$$IF \textit{reason}_1 \textit{ AND } \dots \textit{ AND } \textit{reason}_n \textit{ THEN } \textit{conclusion}. \quad (1)$$

This means that if the reasons are true the conclusion is also true. Basing on this expression (1) a more elaborate example of rule-making can be presented, in which the condition and conclusion are expressed as a combination of logical statements: *IF a takes the x value AND b takes the y value THEN c takes the z value*.

Within article [10] authors presented a model of semantic service specifications in UML notation. For the purpose of the simulation a knowledge base editor has been prepared to introduce the semantic specification of the service. It allows creating an instance of a service model by introducing the values of attributes through a graphical user interface (GUI) in a form of an object "tree", and saving it as a XML file. As the file format for data exchange standard XMI called Metadata Interchange has been chosen to ensure easy integration of the knowledge base with various IT systems:


```

<semanticgis:AtomicService xmi:version="2.0"
xmlns:xmi="http://www.omg.org/XMI"
xmlns:semanticgis="http://www.semanticgis.net/semanticgis
model"
uniqueResourceIdentifier="http://semanticgis.net:8080/geo
server/wms?bbox50.00,10.00,55.00,25.00&styles=&Fo
rmat=jpg&request=GetMap&version=1.1&layers=to
pp:Roads&width=800&height=317&srs=EPSG:432">
<hasNFP resourceTitle="RoadsMap"
resourceAbstract="Service presents roads in a country"
resourceLanguage="en_en"
responsibleOrganisation="TeleAtlas"
publicationDate="2010-12-12T00:00:00.000+0100"
lastRevisionDate="2011-03-06T00:00:00.000+0100">
<keyword>Road</keyword>
<hasQualityOfService executionPrice="7.0"
executionDuration="24.0" reputation="95.0"
spatialResolution="1:10000"/></hasNFP>
<hasCapability spatialServiceCategory="Digital
Cartographic Model" wsStandard="WMS"
resourceLocator="www.semanticgis.net/capability/roads">
<hasGeographicBoundingBox>
<hasMinCoordinates latitude="50.0" longitude="10.0"/>
<hasMaxCoordinates latitude="55.0" longitude="25.0"/>
</hasGeographicBoundingBox>
<hasLayer layerName="RoadsLayer" layerType="Thematic
layer">
<hasArtifact instanceOf="semanticgis.net.ontology.Road"/>
</hasLayer>
</hasCapability>
</semanticgis:AtomicService>.

```

The data structure shown above can be interpreted as a rule of decision-making (1), where the individual attributes of the model and their values such as e.g. *GeographicBoundingBox.MaxCoordinates.latitude="50.0"* etc. are the conditions. Whereas the conclusion is a unique Web services identifier (URI), as a value assigned to the *AtomicService.uniqueResourceIdentifier* attribute. The structure shown above represents a single fact in the knowledge base and is loaded into the working memory of rules engine.

Another type of facts used to represent knowledge are ontological concept instances, creating dictionaries of particular domain. Decision rules built on their basis link information about the concept (e.g., *city*) with its attributes such as *name*, *population*, *geographic location*, etc. They can be used in the process of inference as

elements of queries of Domain Specific Language when the user is looking for some geographical location concepts such as: *city - Zielona Gora, country - Poland*, etc. The listing below shows how to define a decision rule determining a concept instance of a *City (The city of Warsaw)*. In this case, the information is stored in a WSML format [3]:

```
instance Warsaw memberOf City
  name hasValue "Warsaw"
  cityCoordinates hasValue WarsawCoordinates
  population hasValue 1714446
  zipCode hasValue "00-xxx - 05-xxx".
```

This data structure can be interpreted as a decision rule (1), where each object attributes and their values e.g. *City.population = '1714446'* etc. are the conditions. The conclusion is the concept instance identifier (instance *Warsaw member Of City*). Also, this structure represents a single fact in the knowledge base.

The user interface which is a part of the functional construction of the integrated expert system, allows the user to communicate directly with the system using a DSL format similar to natural language. Language grammar based on production rules (2) has been designed by the authors

$$\text{WHEN } condition_1, \dots, condition_n \text{ THEN } action_1, \dots, action_n. \quad (2)$$

Further in this article there are examples of queries to the system designed on the basis of this grammar. The conclusion is that the decision rules (*IF... THEN*) have been used to represent the facts stored in the knowledge base. However, the production rules (*WHEN... THEN*) can be used to submit queries to the expert system in a language specific to the field of GIS. To build the integrated expert system, Drools [1] rules engine based on Rete [5] algorithm, which represents the class of production systems, has been used.

3.1 Experiment Assumptions

The main criterion for the simulation results assessment was correctness of the integrated expert system. This means that the system should provide a high level of expertise to be issued in an acceptable time. Therefore a comprehensive set of production rules in DSL has been prepared, searching for services according to a variety of features.

Compliance of assessed results with the targets was evaluated. An acceptable amount of time to issue an expertise is amount of time which allows searching for Web services distributed on the Internet through the user interface of a Web browser. Therefore, an average time of an expertise delivery by the expert system has also been assessed, when the system base contained several thousands of semantic specifications of services. This article presents examples of the results representative for a wide range of DSL queries. The simulation has been carried out on a computer

equipped with Windows 7 operating system, Intel i7-950 Quad-Core 3.06 GHz Processor, 6 GB DDR3 1600 MHz RAM, virtual Java 1.6 machine.

3.2 Simulation Results

The task of DSL inquiry (Example 1) presented below is to build a complex service presenting a roads and a rivers between two cities:

```

rule "Roads and rivers between cities"
when There is a City1 where City1 name is "Warsaw"
      There is a City2 where City2 name is
"Berlin"
      There is a Service1 where
      - presented artifact by Service1 is "Road"
      - boundaries of Service1 contains City1
      - boundaries of Service1 contains City2
      There is a Service2 where
      - presented artifact by Service2 is
"River"
      - boundaries of Service2 contains City1
      - boundaries of Service2 contains City2

then Propose Services.

```

The next production rules' task (Example 2) is to find specified map layers around the city. In addition, using elements of fuzzy logic a *low execution duration* and an acceptable *average execution price* have been set. Using DSL expressions the expert system finds the best services which meet the defined constraints. Algorithm for decision making in fuzzy environment (maximizing decision) has been implemented by the authors in publication [9] and included within the tested prototype.

```

rule "Available maps around Warsaw"
when There is a City1 where City1 name is "Warsaw"
      There is a Service1 where
      - boundaries of Service1 contains City1
      - service type is "WMS"
      Execution duration should be "low"
      Execution price should be "average"

then Create service options set
rule "Make best decision in fuzzy surroundings"
when There is Service options set
then Propose best services

```

The complexity of the production rule has an impact on the system response time depending on the number of facts stored in the knowledge base.

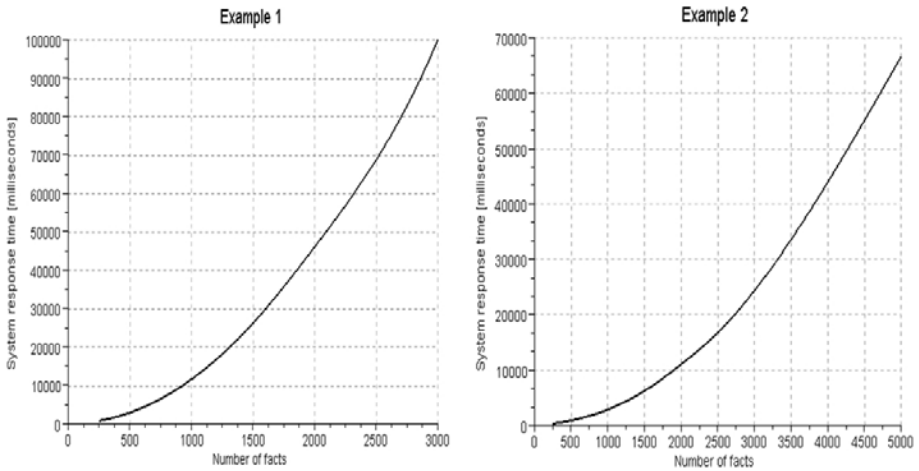


Fig. 2. System response time and amount of facts dependency

Fig. 2 shows a graph of the considered dependency. For Example 1 the response time of 10 seconds was obtained for about 1,000 semantic specification of services (facts) processed by the system. The response time of less than 1 minute was obtained for about 2,250 facts. For inquiry from Example 2, the system response time within 10 seconds was possible for a knowledge base consisting of approximately 2,000 facts. The response time of less than 1 minute was obtained for approximately 4,500 facts.

4 Summary

This paper presents a description of simulation experiment regarding a method for the composition of complex service in the field of Internet Geographic Information Systems. Knowledge base was created on the basis of actual Web services, for which the authors have prepared a semantic specification.

Based on the opinions issued by the developed integrated expert system, it can be stated that it has acceptable correctness. Queries expressed in the DSL were of great diversity, in the context of established to obtain the expertise. Despite the high level of flexibility to formulate these queries, conclusions presented by the system were consistent with the stated objective.

Nowadays, the main criterion posed by Internet users is information time access. Studies have shown that in less than 10 seconds an answer of expert system, with an integrated knowledge base consisting of about one thousand to two thousand facts can be obtained. These results provide a basis for finding that a time criterion can be achieved by using more powerful processors and server sets, or by dividing the search task within distributed systems (e.g., agent systems).

Acknowledgements. The first author is a scholar within Sub-measure 8.2.2 Regional Innovation Strategies, Measure 8.2 Transfer of knowledge, Priority VIII Regional

human resources for the economy Human Capital Operational Programme co-financed by European Social Fund and state budget. Lubuskie - Worth your while.

References

1. Bali, M.: Drools JBoss Rules 5.0 Developer's Guide. Packt Publishing (2009)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American* 284(5), 34–43 (2001)
3. Cimpian, E., Meyer, H., Roman, D.: Ontologies and Matchmaking. In: Kuroпка, D., Troeger, P., Staab, S., Weske, M. (eds.) *Semantic Service Provisioning*, pp. 19–54. Springer, Heidelberg (2008)
4. Davies, J., Studer, R., Warren, P.: *Semantic Web Technologies, trends and research in ontology-based systems*. Wiley, Chichester (2006)
5. Doorenbos, R.B.: Production matching for large learning systems (Rete/UL). Ph.D. thesis, Carnegie Mellon University (1995)
6. Fahringer, T., Krause, H.: Adaptive Services Grid – White Paper: ASG technology advantages and disadvantages, exploitation possibilities and its business impact. In: Fahringer, T., Krause, H. (eds.) *ASG DVD* (2007), <http://www.asg-platform.org>
7. Fensel, D., Lausen, H.: *Enabling Semantic Web Services, The Web Service Modeling Ontology*. Springer, Heidelberg (2007)
8. Fu, P., Sun, J.: *Web GIS: principles and applications*. ESRI Press, Redlands (2011)
9. Grobelny, P., Pieczynski, A.: Semantic reasoning in internet-based geographic information systems. In: 15th International Conference on Soft Computing - MENDEL 2009 Proceedings, pp. 127–132. Brno University of Technology (2009)
10. Grobelny, P.: A Method for Reasoning about Complex Services within Geographic Information Systems. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) *KES-AMSTA 2010. LNCS(LNAD)*, vol. 6070, pp. 132–141. Springer, Heidelberg (2010)
11. Hejmanowska, B.: Data Quality Effect on Risk of Decision Processes Supported by GIS Analyses. Dissertation Monographs, vol. (141). Wydawnictwa AGH, Krakow (2005)
12. Kifer, M., Lausen, G., Wu, J.: Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the Association for Computing Machinery* 42(4), 741–843 (1995)
13. Kuroпка, D., Troeger, P., Staab, S., Weske, M.: *Semantic Service Provisioning*. Springer, Heidelberg (2008)
14. Scharl, A., Tochtermann, K.: *The Geospatial Web, How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society*. Springer-Verlag London Limited, Heidelberg (2007)
15. Sharples, M., Hutchinson, C., Hogg, D., Torrance, S., Young, D.: *Computers and Thought: A Practical Introduction to Artificial Intelligence*. MIT Press, Cambridge (1989)
16. Toma, I., Sapkota, B.: An Evaluation Framework for Discovery in Grid and Web Service Environments. *International Journal on Multiagent and Grid Systems*, Special issue (4) (2006)
17. Yang, G., Kifer, M., Zhao, C.: FLORA-2: A Rule-Based Knowledge Representation and Inference Infrastructure for the Semantic Web. In: Chung, S., Schmidt, D.C. (eds.) *ODBASE 2003. LNCS*, vol. 2888, pp. 671–688. Springer, Heidelberg (2003)

OAuth+UAO: A Distributed Identification Mechanism for Triplestores

Dominik Tomaszuk¹ and Henryk Rybiński²

¹ Institute of Computer Science,
University of Bialystok, Poland
dtomaszuk@ii.uwb.edu.pl

² Institute of Computer Science,
Warsaw University of Technology, Poland
h.rybinski@ii.pw.edu.pl

Abstract. The Semantic Web gives users and applications the ability to access and retrieve decentralized resources which may be stored in triplestores. This paper describes a simple identification protocol dedicated to triplestores which is universal and appropriate for the distributed environment. We propose a mechanism based on the HTTP standard, extended with OAuth Protocol and Semantic Web ontology. One can optionally adopt Transport Layer Security protocol. We present a scalable method that allows user authentication and authorization to triplestores with data integrity and confidentiality. The identification mechanism enables users to access triplestore data without disclosing authentication and authorization data.

Keywords: Semantic Web, triplestore, ontology, Resource Description Framework, authentication, authorization, access control list.

1 Introduction

A triplestore is a purpose-built database for the storage and retrieval of Resource Description Framework (RDF) data. Much like in the case of other databases, one can find and modify data in triplestores via a query language, such as SPARQL Protocol and RDF Query Language [1,2].

An RDF triple consists of a subject, a predicate, and an object. In [3] the meaning of subject, predicate and object is explained. The subject denotes the resource, the predicate means traits or aspects of the resource, and expresses a relationship between the subject and the object. A collection of RDF statements intrinsically represent a labeled, directed multigraph. The nodes are the subjects and objects of their triples.

Following [3], let U be the set of all URI references, B an infinite set of blank nodes, L the set RDF plain literals, and D the set of all RDF typed literals. U , B , L and D are pairwise disjoint. Let $O = U \cup B \cup L \cup D$ and $S = U \cup B$, then $T \subset S \times U \times O$ is set of all RDF triples.

The RDF syntax and semantics could be extended to *named graphs* [4]. The *named graphs* data model is a simple variation of the RDF data model. The

basic idea of the model consists in introducing a graph naming mechanism, with the note by G , the set of graphs which can be seen as $G = 2^T$. A named graph is a pair $ng = (n, g)$ with $n \in U$ (called name) and $g \in G$.

RDF information providers do not have any explicit way to express any intention concerning information security. In the paper we attempt to define the proper functions to equip the RDF information in triplestore with security means, such as authentication, authorization, data integrity and confidentiality.

In the context of triplestores there is a need to provide security means similar to those specific to classical databases. In this paper a new distributed identification mechanism based on OAuth [5] protocol and access control ontology for triplestores is presented. OAuth allows users to share private resources stored on one site with another site without having to hand out their credentials, typically login and password. An access control ontology can define a list of permissions attached to a role.

The paper is constructed as follows. Section 2 is devoted to related work. In Section 3, we propose a solution for RDF information security with the use of the following standards and tools: (1) authentication and data integrity with the OAuth protocol and optionally confidentiality over Transport Layer Security; and (2) authorization with the User Access Ontology approach. In Section 4 the implementation of the proposal is presented. The paper ends with conclusions.

2 Related Works

In the classical databases one can distinguish three main categories of security: user authentication for querying the database results [6,7], an access control approach [8,9], and encrypting datasets [10]. The authentication of query results is divided into subcategories: cryptographic primitives, based on digital signature [6] and Merkle Hash Trees [7]. The access control approach consists of two subcategories: content-based access control [8] and rule-based access control [9].

For semi-structured databases, which are more suited to web applications, the main research concentrates on access control [11,12,13]. In [11] are defined access restrictions directly on the structure and content. Other ones [12] present varying protection granularity levels. The access control policy is proposed by using a 5-tuple of subject, object, privilege, propagation option and signed access decision. In [13] an access control model provides provisional authorization.

On the other hand, in web applications there are database independent and resource oriented solutions, which focus in decentralized manner on the authentication and/or authorization procedures. Good examples are OpenID [14,15] and OAuth Protocol [15,5]. The idea of OpenID consists in assigning a unique ID to a resource, which takes the form of a unique URL, and is managed by an OpenID provider, which handles the authentication procedure. While OpenID is all about using a single identity to sign into many sites, OAuth is about giving access to a resource without sharing identity at all. The OAuth protocol provides authentication, authorization and data integrity. Yet another proposal in this area is provided in [16] by means of Security Assertion Markup Language

(SAML), which is a standard for exchanging authentication and authorization data between the identity providers and service providers. Another proposal, which is extended to access control support, is XACML [17]. It is a language based on XML for security policies and access decisions. XACML provides security administrators to describe an access control policy once, without having to rewrite it numerous times in different application-specific languages. While this approach may be sufficient for triplestores using XML syntax to store, it is not satisfactory for other triplestores.

In the context of Semantic Web, there are also new proposals for authorization solutions. Some of the papers [18,19] define policy-based access control. [18] defines policies that describe subgraphs on which various operations, such as *insert*, *remove*, *update* and *read*, are identified by specifying RDF patterns. The authors define a set of policy rules, enforced by a policy engine to reach the authorization decisions. Unfortunately, they do not discuss the semantics in a formal way. In [19] triples are not annotated with accessibility information, but the enforcement mechanism is query-based. The policy permissions are injected into the query in order to ensure that the triples obtained are only the accessible ones. Unfortunately, the semantics of the policy are not formally defined in [19]. Yet another approach is provided in [20,21], which respect RDF Schema (RDFS) entailments. In [20], the authors discuss how conflicts can be resolved using the RDFS subsumption hierarchies. This proposal, just like in [18], requires instantiating the RDF patterns. In [21] inferences are computed for an RDF dataset without revealing information that might have been explicitly not permitted.

In the context of Semantic Web, the first attempts towards solving security issues were presented in [22,23]. They concentrate on assuming explicit and domain-specific trust ratings. An extension of these ideas is presented in [24,25], where the secure authentication protocol WebID¹ was proposed. The solution enables the building of distributed social networks. WebID uses the Web of Trust mechanism [26], but this does not require signing. The friendship relations are not embedded in the signature. Unfortunately, it is not well suited to triplestores. In particular, it cannot be used for larger triplestore resources. Furthermore, WebID only supports authentication. In contrast, here we do not use Web of Trust. We concentrate on defining mechanisms strictly dedicated to triplestores, preserving the feature of using linked data [27] to publish, share, and connect users' and groups' data stored in triplestores. What is important is that, our approach differs from the idea presented in [24,25] in that it does not depend on other users for trust. In addition, WebID uses Transport Layer Security with X.509 certificates which make this form of communication slower and more complicated than our proposal.

3 Distributed Identification Mechanism for Triplestores

In this Section we discuss the idea of using the identification mechanism, as provided by OAuth. Additionally we define a new ontology, devoted to the spec-

¹ WebID is also known as FOAF+SSL.

ification of access control by means of authorization. For the cases where confidentiality is needed, optional, encryption based on Transport Layer Security (TLS) can be used.

3.1 Authentication and Data Integrity over OAuth

In this Section we suggest using OAuth to access a triplestore. It is token-based authentication. That means that a logged-in user has a unique token used to access data from the triplestore. Users access to triplestore data with sharing tokens and without disclosing any identity data. This approach presents a triplestore as a server. It could be, optionally, an authorization endpoint and/or an access control lists repository. A client is an application that uses OAuth to access the triplestore on behalf of the user.

We propose an authorization algorithm using OAuth over Hypertext Transfer Protocol. It consists of the following seven steps:

1. Obtain request token from the triplestore² to the client.
2. Redirect client to the authorization endpoint.
3. The server requests user to sign in by using login and password. It is important that in this step login, and password should be encrypted.
4. If login and password are correct, the server associates the user with the role and asks for approval granting to triplestore.
5. Redirect from the authorization endpoint to the client.
6. Exchange request token for access token.
7. The client is ready to request the private data to triplestore.

OAuth take care of the data integrity by signing HTTP requests. The OAuth protocol is secure, because it has tokens (and the fields: *timestamp* and *nonce*) that do not pass login and password, for verifying unique requests. Each token grants access for a specific triplestore resources and for a defined duration.

The main disadvantage of OAuth is that login, password and other settings, such as email, cannot be changed via this protocol.

OAuth over HTTP does not provide confidentiality. The solution to this problem is to use Hypertext Transfer Protocol Secure (HTTPS). This proposal does not force the use of HTTPS and allows using HTTP, when confidentiality is not needed.

The main disadvantage is that HTTPS requires both parties to the communication to do extra work in exchanging handshakes and encrypting and decrypting the messages, making this form of communication slower than it would be without it.

3.2 Authorization: User Access Ontology

In this Section the proposed User Access Ontology is presented (in the sequel denoted by UAO). UAO is an ontology describing roles, their permissions, and

² In OAuth it is called service provider.

allowed or permitted actions on triplestores. It allows the description of access control lists for users, without assigning them to a single triplestore and/or other databases. UAO is a descriptive vocabulary expressed in Resource Description Framework (RDF), RDF Schema and Web Ontology Language (OWL). The presented ontology is written in the RDF/XML and Terse RDF Triple Language syntaxes [29].

We define an authorization $a \in AuthZ$ as a tuple of the form $\langle role, action \rangle$ where $role \in R$ and $action \in ACT$. The user description (**User** class) consists of first name (**firstName** property), last name (**lastName** property) and user name (**userName** property). The most important is user name, because it identifies and associate the authenticated user with access control lists. The user characteristics can be extended to other ontologies, such as FOAF [28]. Users are assigned (**hasRole** property) to roles (**Role** class). Users should have a minimum of one role. Roles may have names (**roleName** property). There is also default policy for the role (**DefaultPolicy** class), which could deny (**Deny** class) or permit (**Permit** class) access to data. It should have exactly one default policy. Roles are assigned (**hasPermission** property) to their permissions (**Permission** class). Let P be the permissions, USR be the user and DP be the default policy, then role $R \in usr, p, dp$ with $usr \in USR, p \in P$ and $dp \in DP$. Permissions may have numeric priorities (**priority** property), which prevents conflicts. It should also have filters (**filter** property), which are sets of triple pattern TP or URI references U (see Section 1). Let V be the set of all variables, then $TP \subset (S \cup V) \times (U \cup V) \times (O \cup V)$ and V is infinite and disjoint from U, B, L and D (see Section 1). Permissions may have named graph declaration (**graph** property). When this value is not set, the permissions refer to the default graph.

The permissions are assigned (**hasAction** property) to actions (**Action** class). The actions ACT specify roles which are granted to access the triplestore, as well as what operations are allowed or forbidden on the triplestore. We propose nineteen types of actions, which are based on the SPARQL clauses [12] and that can be combined with each other. These types of permission are divided into three groups: graph management (**GraphManage** class), graph modification (**GraphModify** class) and query forms (**QueryFrom** class). The graph management permissions allow the execution of the clauses: CREATE and DROP. The graph modification permissions allow the execution of the clauses: INSERT DATA, INSERT, LOAD, DELETE DATA, DELETE, DELETE WHERE, CLEAR and DELETE/INSERT. The query form permissions allow the execution of the read-only query forms: SELECT, CONSTRUCT, ASK and DESCRIBE. All SPARQL clauses are reflected in classes and presented in Table 1.

Prohibit or permit action names are identical with classes (see Table 1). All actions have own parameters, which are identical to filter values. All actions with parameters are presented in Table 2.

Listing presents the example of access control list in Terse RDF Triple Language. This information is stored in an ACL repository and it could be part of the triplestore. The User Access Ontology is presented in Fig. 1.

Table 1. classes reflected in SPARQL

Class	Superclass	SPARQL clauses
GraphManage	Action	CREATE, DROP
Create	GraphManage	CREATE
Drop	GraphManage	DROP
GraphModify	Action	INSERT [DATA], LOAD, CLEAR, DELETE [DATA] [WHERE], DELETE/INSERT
DeleteInsert	GraphModify	DELETE/INSERT
Load	GraphModify	LOAD
Clear	GraphModify	CLEAR
Add	GraphModify	INSERT DATA. INSERT
Remove	GraphModify	DELETE DATA, DELETE, DELETE WHERE
InsertData	Add	INSERT DATA
Insert	Add	INSERT
DeleteData	Remove	DELETE DATA
Delete	Remove	DELETE
DeleteWhere	Remove	DELETE WHERE
QueryFrom	Action	ASK, DESCRIBE, SELECT, CONSTRUCT
Ask	QueryFrom	ASK
Describe	QueryFrom	DESCRIBE
Select	QueryFrom	SELECT
Construct	QueryFrom	CONSTRUCT

Table 2. Actions with parameters

Class	Action with parameters
GraphManage	GraphManage(ng) with $ng \in U$
Create	Create(ng) with $ng \in U$
Drop	Drop(ng) with $ng \in U$
GraphModify	GraphModify(ng, u, tp_1, tp_2) with $ng \in U, u \in U,$ $\{tp_1 : tp_1 \in TP\}, \{tp_2 : tp_2 \in TP\}$
DeleteInsert	DeleteInsert(ng, tp_1, tp_2) with $ng \in U, \{tp_1, tp_2 : tp_1, tp_2 \in TP\}$
Load	Load(ng, u) with $ng \in U, u \in U$
Clear	Clear(ng) with $ng \in U$
Add	Add(ng, tp) with $ng \in U, \{tp : tp \in TP\}$
Remove	Remove(ng, tp) with $ng \in U, \{tp : tp \in TP\}$
InsertData	InsertData(ng, tp) with $ng \in U, \{tp : tp \in TP\}$
Insert	Insert(ng, tp) with $ng \in U, \{tp : tp \in TP\}$
DeleteData	DeleteData(ng, tp) with $ng \in U, \{tp : tp \in TP\}$
Delete	Delete(ng, tp) with $ng \in U, \{tp : tp \in TP\}$
DeleteWhere	DeleteWhere(ng, tp) with $ng \in U, \{tp : tp \in TP\}$
QueryFrom	QueryFrom(ng, tp) with $\{ng : ng \in U\}, \{tp : tp \in TP\}$
Ask	Ask(ng, tp) with $\{ng : ng \in U\}, \{tp : tp \in TP\}$
Describe	Describe(ng, tp) with $\{ng : ng \in U\}, \{tp : tp \in TP\}$
Select	Select(ng, tp) with $\{ng : ng \in U\}, \{tp : tp \in TP\}$
Construct	Construct(ng, tp) with $\{ng : ng \in U\}, \{tp : tp \in TP\}$

Example of access control list

```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix uao: <http://example.org/uao#> .

_:u01 a uao:User ;
    uao:firstName "John" ;
    uao:lastName "Smith" ;
    uao:userName <http://example.org/card#me> ;
    uao:hasRole _:r01 .

_:r01 a uao:Role ;
    uao:roleName "teachers" ;
    uao:hasDefaultPolicy uao:Permit ;
    uao:hasPermission _:p01 .

_:p01 a uao:Permission ;
    uao:priority "10"^^xsd:int ;
    uao:graph "$g" ;
    uao:filter "($s $p $o)" ;
    uao:hasAction uao:Select .
    
```

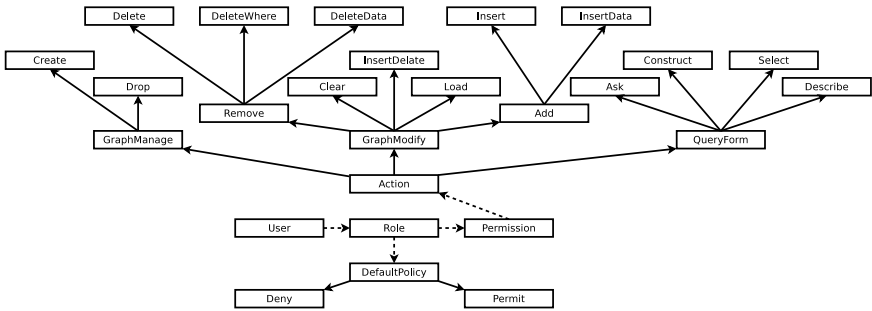


Fig. 1. User Access Ontology

4 Implementation

Now we present the implementation of our approach. We used PHP5 as the development platform. The system consists of the following five parts: query engine, access control lists repository (applying User Access Ontology), protected resources (a part of triplestore), authorization endpoint and HTTP client. Our implementation of the proposed prototype is made of three internal modules, namely query engine, access control lists repository and protected resources. Fig. 2 presents the system architecture.

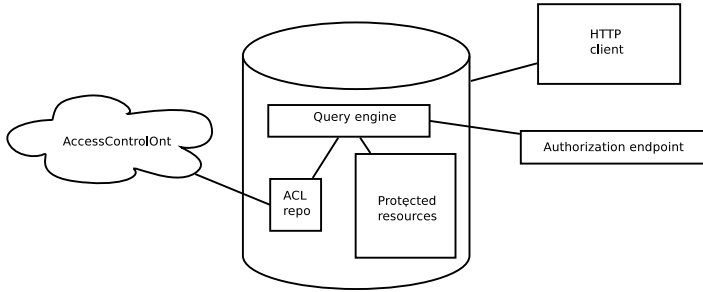


Fig. 2. System architecture

Our modules are additional layers on top of a document-oriented database (MongoDB), which substitutes the triplestore. The access control lists and protected resources are stored in triplestore. We use external, third-party OAuth authorization endpoint and a web browser as an HTTP client to test the prototype. The UML sequence diagram (Fig. 3) presents the workflow of our implementation. The diagram shows basic stages of interaction between actors:

1. The web browser obtains an unauthorized request token.
2. The authorization point and access control list repository authorizes the request token.
3. The web browser exchanges the request token for an access token.
4. The user executes SPARQL queries.

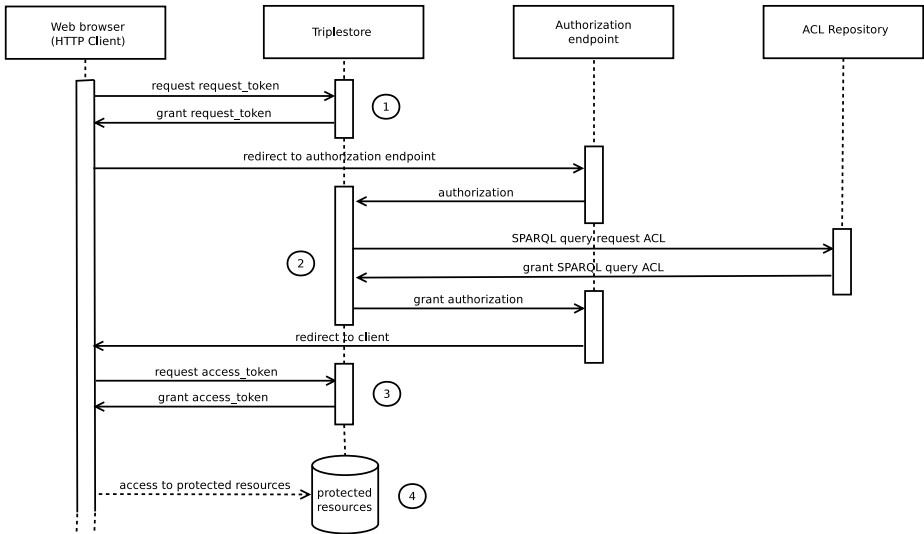


Fig. 3. UML sequence diagram

To enforce an access control, we analyze SPARQL queries and compare to permit or deny actions as exceptions to default policy. Next, the triple patterns from `filter` and/or `graph` properties are mapped to the relevant clauses of SPARQL queries. If this wildcard is true for user permissions, the query is executed.

5 Conclusions

The problem of how to adjust access control to a triplestore has produced many proposals. Most of them are hard to use without dedicated tools, hence making the problem seem difficult and slow. We assume that the triplestores, to be more functional, should provide a mechanism to confirm the identity of a user. This mechanism also confirms restrictions that operate on the RDF graphs. The main motivation for this paper is a lack of such requirements.

We have produced a simple, thought-out RDF standard based and closed triplestores proposal. We believe that our idea is an interesting approach, because it is triplestore independent. We have proposed an identification protocol dedicated to triplestores that is universal and distributed. It uses HTTP, OAuth and ontology. Our proposal can work either with mobile and other devices or a web browser and other software as a triplestore client and server. Another advantage is that users do not have to give a password to third parties. A crucial advantage of the proposal is that the identification mechanism is distributed and adopts linked data. The implementation shows its great potential.

We realize that some further work on this issue is still necessary. We are currently investigating our proposal to support web SPARQL endpoint and RESTful managing RDF graphs.

References

1. Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF, World Wide Web Consortium (2008)
2. Schenk, S., Gearon P., Passant A.: SPARQL 1.1 Update, World Wide Web Consortium (2010)
3. Klyne, G., Carroll, J.J.: Resource Description Framework (RDF): Concepts and Abstract Syntax. World Wide Web Consortium (2004)
4. Carroll, J.J., Bizer, C., Hayes, P., Stickler, P.: Named graphs, provenance and trust. In: 14th International Conference on World Wide Web. ACM, New York (2005)
5. Hammer-Lahav, E.: The OAuth 1.0 Protocol. Internet Engineering Task Force (2010)
6. Mykletun, E., Narasimha, M., Tsudik, G.: Authentication and integrity in outsourced databases. *ACM Transactions on Storage* (2006)
7. Li, F., Hadjieleftheriou, M., Kollios, G., Reyzin, L.: Dynamic authenticated index structures for outsourced databases. In: Special Interest Group on Management Of Data. ACM, New York (2006)
8. Bertino, E., Haas, L.M.: Views and security in distributed database management systems. In: Schmidt, J.W., Missikoff, M., Ceri, S. (eds.) *EDBT 1988*. LNCS, vol. 303, pp. 155–169. Springer, Heidelberg (1988)
9. Ahn, G., Sandhu, R.: Role-based authorization constraints specification. *ACM Transactions on Information and System Security, TISSEC* (2000)

10. Hacigümüş, H., Iyer, B., Li, C., Mehrotra, S.: Efficient execution of aggregation queries over encrypted relational databases. *Database systems for Advanced Applications* (2004)
11. Paraboschi, S., Samarati, P.: Regarding access to semistructured information on the web. In: 16th IFIP TC11 Annual Working Conference on Information Security: Information Security for Global Information Infrastruktüre (2000)
12. Bertino, E., Castano, S., Ferrari, E., Mesiti, M.: Controlled access and dissemination of XML documents. In: *WIDM 1999 Proceedings of the 2nd International Workshop on Web Information and Data Management*. ACM, New York (1999)
13. Jajodia, S., Kudo, M., Subrahmanian, V.S.: Provisional authorizations. *E-commerce Security and Privacy* (2001)
14. Recordon, D., Reed, D.: OpenID 2.0: a platform for user-centric identity management. In: *The Second ACM Workshop on Digital Identity Management*. ACM, New York (2006)
15. Kaila, P.: OAuth and OpenID 2.0. *The Seminar on network security* (2008)
16. Cantor, S., Kemp, J., Philpott, R., Maler, E.: Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. *Organization for the Advancement of Structured Information Standard* (2005)
17. Moses, T.: eXtensible Access Control Markup Language (XACML) Version 2.0, *Organization for the Advancement of Structured Information Standard* (2005)
18. Reddivari, P., Finin, T., Joshi, A.: Policy based Access Control for a RDF Store. In: *Proceedings of the Policy Management for the Web Workshop* (2005)
19. Abel, F., Luca De Coi, J., Henze, N., Koesling, A.W., Krause, D., Olmedilla, D.: Enabling advanced and context-dependent access control in RDF stores. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) *ASWC 2007 and ISWC 2007*. LNCS, vol. 4825, pp. 1–14. Springer, Heidelberg (2007)
20. Kim, J., Jung, K., Park, S.: An introduction to authorization conflict problem in RDF access control. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) *KES 2008, Part II*. LNCS (LNAI), vol. 5178, pp. 583–592. Springer, Heidelberg (2008)
21. Jain, A., Farkas, C.: Secure resource description framework: an access control model. In: *11th ACM Symposium on Access Control Models and Technologies*. ACM, New York (2006)
22. Golbeck, J., Parsia, B., Hendler, J.: Trust networks on the semantic web. In: Klusch, M., Omicini, A., Ossowski, S., Laamanen, H. (eds.) *CIA 2003*. LNCS(LNAI), vol. 2782, pp. 238–249. Springer, Heidelberg (2003)
23. Richardson, M., Agrawal, R., Domingos, P.: Trust management for the semantic web. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) *ISWC 2003*. LNCS, vol. 2870, pp. 351–368. Springer, Heidelberg (2003)
24. Story, H., Harbulot, B., Jacobi, I., Jones, M.: FOAF+SSL: RESTful Authentication for the Social Web. In: *European Semantic Web Conference* (2009)
25. Gamble, M., Goble, C.: Standing on the Shoulders of the Trusted Web: Trust, Scholarship and Linked Data. In: *Web Science Conference* (2010)
26. Khare, R., Rifkin, A.: Weaving a Web of trust. *World Wide Web Journal - Special issue: Web security: a matter of trust* (1997)
27. Bizer, C., Heath, T., Berners-Lee, T.: *Linked Data - The Story So Far*. *International Journal on Semantic Web and Information Systems* (2009)
28. Brickley, D., Miller, L.: *FOAF Vocabulary Specification 0.98*. FOAF Project (2010)
29. Beckett, D., Berners-Lee, T.: Turtle - Terse RDF Triple Language. In: *World Wide Web Consortium* (2008)

Propagating and Aggregating Trust with Uncertainty Measure

Anna Stachowiak

Faculty of Mathematics and Computer Science
Adam Mickiewicz University
Umultowska 87, 61-614 Poznań, Poland
aniap@amu.edu.pl

Abstract. Trust networks have been recognized as a valuable component of many modern systems, such as e-commerce or recommender systems, as they provide a way of quality assessment.

In addition to adequate modeling of trust in such network, two fundamental issues need to be addressed: the methods of propagation and aggregation of trust.

In this paper we present an operator that performs both propagation and aggregation of trust. Trust is modeled on the basis of IFS theory (Atanassov's intuitionistic fuzzy set theory) with particular emphasis on uncertainty, and the operator is based on relative scalar cardinality of IFS. The operator can be used in a very flexible manner for prediction of local and global trust.

Keywords: Trust aggregation, trust propagation, intuitionistic fuzzy sets, IFS, relative scalar cardinality of IFS, local trust, global trust.

1 Introduction

Widespread access to the Internet caused a growth of networks that are large as never before, with numerous users sharing common interests or business. In such a large society relationships between users may be of different quality, there is a risk of abuse or malicious intrusion. It is thus significant to introduce some quality measure for relations, and one of such measure is trust that is expressed by one user towards another ([5]).

Many methods of modeling trust were proposed in the literature ([3], [4], [2]). As stated in the article of De Cock et al.([2]), modeling a trust score in a dual form of trust and distrust (t, d) benefits in many ways, among others enables to distinguish the situation of lack of trust $(0, 1)$ from the one of lack of knowledge $(0, 0)$. What is more, representing trust and distrust by any number from the interval $[0,1]$ enables to express such imprecise attitudes like "to trust somebody very much" or "to rather not trust somebody".

In general, trust and distrust do not have to sum up to 1, however in this paper we assume that they satisfy the condition $t + d \leq 1$, which is the main condition in the intuitionistic fuzzy set theory (IFS theory). Without this restriction we have an interesting bilattice-based model that allows inconsistency (see [9]).

The details concerning trust modeling are presented in Section 2. We will focus especially on an uncertainty value and will emphasize the informativeness of this factor.

Next we take advantage of some of the aspects of IFS theory when constructing an operator of trust propagation and aggregation. The usual approach to this problem is to use two separate operators for propagation and aggregation. We propose a unified method based on a simple idea of counting elements. Some properties of this operator, illustrated with examples, are the subject of Section 3.

2 Basic Concepts of Trust Network

Let us consider a network of users (people, agents, peers, robots etc.) where there is no central certification mechanism, but the only way to evaluate some user reliability is with the use of subjective opinions of other users. In other words, information about reliability is incorporated into a structure named trust network or web of trust.

2.1 Trust Modeling

A trust network is a triplet (V, E, R) where V is a set of users and E is a set of directed trust links between them. R is an $E \rightarrow [0, 1]^2$ mapping that for each pair of users (a, b) associates a trust score $R(a, b) = (t, d)$, where t is called the trust degree of a in b and d is called the distrust degree.

We model a trust score using Atanassov's IFS theory ([1]). An IFS \mathcal{E} is a pair:

$$\mathcal{E} = (A^+, A^-),$$

where A^+ is a fuzzy set of elements that belong to \mathcal{E} , and A^- is a fuzzy set of elements that do not belong to \mathcal{E} . This theory, in contrast with fuzzy set theory, incorporates uncertainty about the membership of an element, as A^- is not necessarily a negation of A^+ , but $A^- \subset (A^+)^c$ where the complement of fuzzy set A is defined as $A^c(x) = 1 - A(x)$ for each x . Therefore, the value $1 - A^+(x) - A^-(x)$ reflects uncertainty or hesitation about membership of an element x in IFS \mathcal{E} . Similarly, for example due to lack of knowledge, uncertainty (u) is present when specifying trust and distrust degrees, and is equal to:

$$u = 1 - t - d. \quad (1)$$

It is a difficult problem how to interpret and operate on distrust. In this paper distrust only forms the boundary that the trust could not exceed, and a special emphasis is put on uncertainty. Semantics of uncertainty is rich:

- $(0, 0)$ - uncertainty value is maximal and equals 1; it represents the situation of total confusion;
- $(1, 0)$ or $(0, 1)$ - uncertainty is 0, the knowledge is full;
- $(0.5, 0.5)$ - uncertainty is also 0, but this time we lack knowledge because opinions about trust were equally distributed between "yes" and "no".

Despite equal uncertainty values the third case is significantly different from the second one. To distinguish those two situations we propose to use a general approach to IFS theory, described in [6], where uncertainty degree is t-norm dependent:

$$u = \nu(t) T \nu(d) \tag{2}$$

where T is a t-norm, e.g. algebraic t-norm: $aT_a b = a \cdot b$ or Łukasiewicz t-norm: $aT_L b = 0 \vee (a + b - 1)$, and ν is a strong negation.

If we take e.g. algebraic t-norm, then for (0.5, 0.5) we obtain uncertainty equal to 0.25.

2.2 Trust Propagation and Aggregation

In the environment of trust network there exist two possible ways of predicting trust scores. First one is *local* (or subjective) - this is a reliability of a user from the point of view of another user. Second option is *global* (or objective). It is one global value often called the reputation of a user that represents what the community as a whole (on "average") think about this user.

Obviously, in a big network it is impossible for a user to express its trust score against every other user. Instead, we use a mechanism of *trust propagation* to estimate it. We assume transitivity - if a user a trusts (to some degree) in user b and user b trust in c , then we can somehow estimate the trust score of a in c . In general, using propagation we are able to calculate a (local) trust score of a in some z if there is a path connecting users a and z in a trust network. If there is more then one trust path between users, then we have to perform *trust aggregation*.

3 Operator of Trust Propagation and Aggregation

In this paper we propose an operator that combines both trust propagation and aggregation functionality. It is based on a simple observation: if most of our friends trust somebody, we would also be willing to trust that person. Thus, if we need local trust of user x in y , a question can be formulated:

"How many of users trusted by x trust y ?"

Similarly, if a global trust score of user y is to be found, we can ask:

"How many users trust y ?"

Notice that both of this questions are questions about relative cardinality. We describe it in more details in the following subsections.

3.1 Local Trust

The basic concept of estimating local trust was presented in [7]. In this paper we want to emphasize the aggregating functionality of this operator. As mentioned, if trust score of x in y is to be found we would ask:

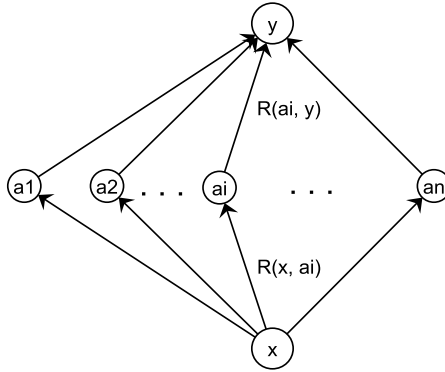


Fig. 1. A model of trust network

”How many of users trusted by x trust y ?” (Q1)

Consider a simple case, depicted in Fig.1, when the path length between x and y is equal 2. We construct two IFSs: ”users trusted by x ” and ”users that trust y ”. Using notation from Fig.1. we define those sets as:

- $X : \{R(x, a_i) | i = 1, \dots, n\}$ - an IFS of users a_1, \dots, a_n trusted by x to a degree $R(x, a_1), \dots, R(x, a_n)$;
- $Y : \{R(a_i, y) | i = 1, \dots, n\}$ - an IFS of users a_1, \dots, a_n that trust y to a degree $R(a_1, y), \dots, R(a_n, y)$.

The question Q1 is thus the question about relative scalar cardinality of two IFSs X and Y :

$$\sigma_I(Y|X) = \frac{\sigma_I(Y \cap_{T,S} X)}{\sigma_I(X)}. \tag{3}$$

The intersection of two IFSs $\mathcal{E} = (A^+, A^-)$ and $\mathcal{F} = (B^+, B^-)$ is defined using t-norm T and t-conorm S as:

$$\mathcal{E} \cap_{T,S} \mathcal{F} = (A^+ \cap_T B^+, A^- \cup_S B^-).$$

Scalar cardinality of IFS \mathcal{E} is equal to:

$$\sigma_I(\mathcal{E}) = [\sigma_f(A^+), \sigma_f((A^-)^c)],$$

where

$$\sigma_f(A) = \sum_{x \in \text{supp}(A)} f(A(x))$$

is a scalar cardinality of fuzzy set a with f being a non-decreasing function $f : [0, 1] \rightarrow [0, 1]$ such that $f(0) = 0, f(1) = 1$ called cardinality pattern. Thus, the relative scalar cardinality of IFS from (3) is a proportion of two intervals:

$$\sigma_I(Y|X) = \frac{[\sigma_f(Y^+ \cap_T X^+), \sigma_f((Y^- \cup_S X^-)^c)]}{[\sigma_f(X^+), \sigma_f((X^-)^c)]}.$$

In [7] we proposed to simplify this expression by making some assumptions that are however to strict in some cases. That is why we propose to calculate a local trust simply as:

$$LT(X, Y) = \left(\frac{\sigma_f(Y^+ \cap_T X^+)}{\sigma_f((X^-)^c)}, 1 - \min \left(1, \frac{\sigma_f((Y^-)^c \cap_T (X^-)^c)}{\sigma_f(X^+)} \right) \right).$$

When the distance between x and y is greater then 2 then we execute LT recursively. Notice that a single operation only requires knowledge about the direct neighbour of a given node. We request the information about the x 's neighbours' trust in y . If any of them doesn't know the answer then he propagate the question further by asking his friends, and so on. This recursive procedure was described in [7]. We will use the notation LT_p to denote that we propagate trust along path of length p .

LT is in fact a family of operators. When using algebraic t-norm T_a and identity function as a cardinality pattern it behaves like arithmetic mean weighed with trust scores of the source user. Choosing different t-norms and cardinality patterns gives much more flexibility, e.g we can set up a threshold of minimum trust to be taken into account. It is not desirable to use t-norm minimum, as it preserves information only about one component.

The presented operator has some other intuitive properties, both for propagation and aggregation.

As for propagation, it seems to be reasonable that the longer the path from x to y , the bigger the uncertainty about trust. Indeed, increasing the length of the path from x to y causes increase of uncertainty when using the presented method. It is therefore not meaningful to consider long path when propagating trust, because at some point uncertainty grows beyond acceptable value. Also from computational point of view it is desirable to restrict the path length.

On the other hand, introducing additional information when aggregating trust - adding a new, relatively certain, trust path from x to y - results in decreasing uncertainty. To be more specific, the operator fulfills two postulates for aggregation operators defined in [10] - trust and distrust boundaries. The third condition of knowledge preservation is also fulfilled if we think about uncertainty value as of a piece of information, which is the intention of presented solution.

Those properties are illustrated in the example below.

Example 1. Let us consider small subparts of trust network from Fig.2a)-d). In the calculations we will use identity function as a cardinality pattern and two t-norms: algebraic and Łukasiewicz.

In Fig.2a) the user a has two friends - b and c , that are in turn friends with d . We want to estimate trust score of a in d taking into account two trust paths:

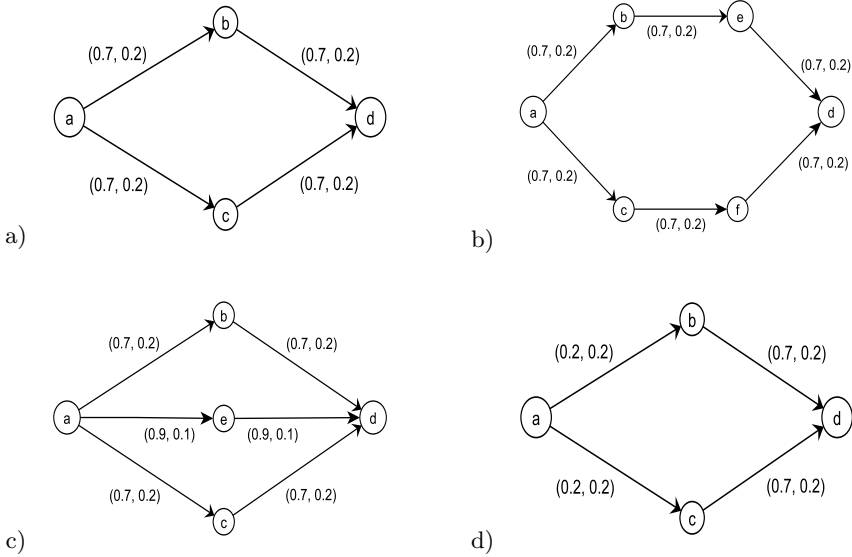


Fig. 2. Fragments of trust networks for Example 1

$a - b - d$ and $a - c - d$. All trust scores which we base on are accompanied by an uncertainty of 0.1. After applying the operator $LT(A, D)$ we obtain:

- (a) $T = T_a: LT(A, D) = (0.61, 0.09)$ with uncertainty = 0.3
- (b) $T = T_L: LT(A, D) = (0.5, 0.14)$ with uncertainty = 0.36

We can see that the uncertainty margin has increased, and will increase more if we lengthen the path from a to d , as can be seen on Fig.2b). The results are presented below:

- (a) $T = T_a: LT(A, D) = (0.54, 0)$ with uncertainty = 0.46
- (b) $T = T_L: LT(A, D) = (0.25, 0.06)$ with uncertainty = 0.69

On the other hand, adding a new (more certain) path to network, like in Fig.2c), results in decreasing uncertainty:

- (a) $T = T_a: LT(A, D) = (0.72, 0.09)$ with uncertainty = 0.19
- (b) $T = T_L: LT(A, D) = (0.64, 0.13)$ with uncertainty = 0.23

Furthermore, using nilpotent t-norm like Łukasiewicz causes that small values of trust won't influence the final result. Similarly, by choosing an appropriate cardinality pattern we may set a threshold of minimal trust score. The sample network is depicted in Fig.2d) and the results are:

- (a) $T = T_a: LT_2(A, D) = (0.175, 0)$ with uncertainty = 0.825
- (b) $T = T_L: LT_2(A, D) = (0, 0)$ with uncertainty = 1

(c) $f = f_{p=0.5}$ and $T = T_a$: $LT_2(A, D) = (0, 0)$ with uncertainty = 1

where:

$$f_p(x) = \begin{cases} \frac{x-p}{1-p}, & \text{if } x > p, \\ 0, & \text{otherwise} \end{cases}$$

3.2 Global Trust

A special case of the question Q1 is a question:

”How many users trust y ?” (Q2)

This question can be used to estimate a general trust in a user, an objective (aggregated) value of reliability of a user in a whole network. Similarly to Q1, we compute a relative scalar cardinality of an IFS of users that trust user y , but relatively to the set of all other users. We denote:

- $Y : \{R(a_i, y) | i = 1, \dots, n\}$ - an IFS of users a_1, \dots, a_n that trust y to a degree $R(a_1, y), \dots, R(a_n, y)$,
- M - a set of all users in the network except y .

Then,

$$\sigma_I(Y|1_M) = \frac{1}{|M|} \sigma_I(Y). \tag{4}$$

Global trust is thus an IFS-value equals to:

$$GT(Y) = \left(\frac{\sigma_f(Y^+)}{|M|}, 1 - \frac{\sigma_f((Y^-)^c)}{|M|} \right).$$

We assume that those users that did not express their trust in y , trust it to a degree $(0,0)$. However, it can be changed. Notice that this approach may be enriched with using previously presented LT . Namely, besides calculating global trust only on the basis of directly connected users, we may predict trust scores in y of distant users. It causes that the opinion of more trusted users would be multiplied, i.e. it would have greater influence on the final global trust. In unique circumstances all possible trust connections may be first calculated, but obviously in a big network it would be computationally expensive. Again, it seems enough to propagate trust along paths of 2 or 3 nodes long. We will use the notation GT_p to indicate that the process of calculating global trust was preceded by propagating trust along path of length p using LT_p .

Nevertheless, the situation when we are able to estimate trust scores of every node in y is interesting - global trust of y may be then understood as a local trust of some hypothetical supervisor (*super*) in y . We assume that such object trusts all users equally (except y) in the network. Then holds:

$$GT_{p=max}(X) = LT_{p=max}(Super, X).$$

Those ideas are demonstrated in the example below.

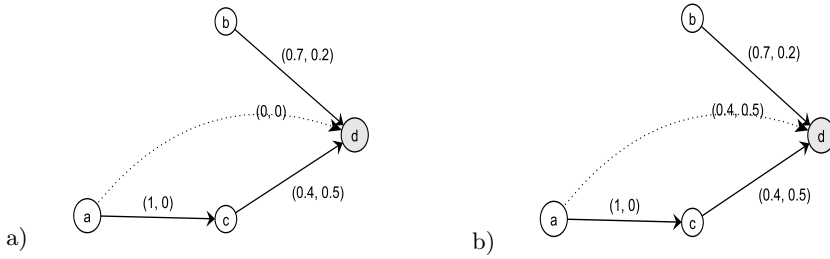


Fig. 3. Trust networks for Example 2

Example 2. Consider a small network, consisting of 4 nodes, depicted in Fig.3. First, we compute $GT(D)$, that takes into account only trust scores of directly connected nodes. Thus, for $f = id$ we get:

(a) $GT_1(D) = (0.37, 0.23)$ with uncertainty = 0.4.

In this case we assume that trust score of a in d equals $(0, 0)$.

If we want to take into account also a trust score of a in d we may estimate it by calculating $LT_p(A, D)$ using $f = id$ and $T = T_a$. Because node c is trusted by a , now the opinion of c is more influential on the final result. Namely,

(b) $GT_2(D) = (0.5, 0.4)$ with uncertainty = 0.1.

Notice, that

$$LT_3(Super, D) = (0.5, 0.4) = GT_2(D).$$

4 Conclusions and Further Work

When modeling trust score with only one value - trust - we lack information about trust provenance and a quality of this information. In this paper we have stated that an uncertainty factor can play a significant role in solving this problem. Uncertainty linked with trust and distrust can be conveniently modeled with IFS theory, that is why we have chosen this tool. Uncertainty gives us additional information about the "quality" of the opinion about trust - if it is certain or vitiated by ignorance. It is even possible to distinguish the case of ambivalent opinions if we use general IFS theory extended with t-norms. What is more, uncertainty value allows to reflect the impact of the propagation path length on the final trust score.

In such environment we have proposed an operator that forms a unified method for propagation and aggregation of trust, allowing estimation of both local and global trust. We have considered some properties of this operator, focusing on its influence on uncertainty factor. The additional merit of the presented operator is the possibility of setting parameters - t-norm and cardinality pattern - to adopt its behavior to particular problems.

Although the idea of counting elements is very simple, it seems that it gives a powerful tool. The aggregation and propagation operation depends only on the information acquired from adjacent neighbors which makes it scalable. This conclusion may initiate a further research consisting in experiments on a large model.

Acknowledgment. The research has been partially supported by the Ministry of Science and Higher Education Grant N N519 384936.

References

1. Atanassov, K.: *Intuitionistic Fuzzy Sets. Theory and Applications* (1999)
2. De Cock, M., Pinheiro da Silva, P.: A Many Valued Representation and Propagation of Trust and Distrust. In: Bloch, I., Petrosino, A., Tettamanzi, A.G.B. (eds.) *WILF 2005. LNCS (LNAI)*, vol. 3849, pp. 114–120. Springer, Heidelberg (2006)
3. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of Trust and Distrust. In: *Proc. WWW 2004*, pp. 403–412 (2004)
4. Josang, A., Knapskog, S.J.: A Metric for Trusted Systems. In: *Proc. NIST-NCSC*, pp. 16–29 (1998)
5. Massa, P., Avesani, P.: Trust metrics in recommender systems. In: *Computing with Social Trust*, pp. 259–285. Springer, Heidelberg (2009)
6. Pankowska, A., Wygalak, M.: General IF-sets with triangular norms and their applications to group decision making. *Information Sciences* 176(18), 2713–2754 (2006)
7. Stachowiak, A.: Trust propagation based on group opinion. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) *IPMU 2010. Communications in Computer and Information Science*, vol. 80, pp. 601–610. Springer, Heidelberg (2010)
8. Wygalak, M.: *Cardinalities of Fuzzy Sets*. Springer, Heidelberg (2003)
9. Victor, P., Cornelis, C., De Cock, M., Pinheiro da Silva, P.: Gradual trust and distrust in recommender systems. *Fuzzy Sets and Systems* 160(10), 1367–1382 (2009)
10. Victor, P., Cornelis, C., De Cock, M., Herrera-Viedma, E.: Bilattice-Based Aggregation Operators for Gradual Trust and Distrust. In: *Proceedings of the International FLINS Conference on Foundations and Applications of Computational Intelligence*, pp. 505–510 (2010)
11. Zadeh, L.A.: Fuzzy sets. *Inform. and Control* 8, 338–353 (1965)

On Ordered Weighted Reference Point Model for Multi-attribute Procurement Auctions

Bartosz Kozłowski and Włodzimierz Ogryczak

Warsaw University of Technology,
Institute of Control & Computation Engineering, 00-665 Warsaw, Poland
{b.kozlowski,w.ogryczak}@elka.pw.edu.pl

Abstract. Multi-attribute auctions (also called multidimensional auctions) facilitate negotiations based on multiple attributes, thus escape from the standard price-only domain into a rich multidimensional domain that can comprise additional attributes like e.g. guarantee conditions or quality. Most multi-attribute preference models used in auction mechanisms are based on a ranking derived from weighted sum. Recently, the Reference Point Method (RPM) approach has been applied to express the multi-attribute preference models within the auction mechanisms allowing to overcome the weighted sum drawbacks. The Ordered Weighted RPM model enables us to introduce importance weights to affect achievements importance by rescaling their measures accordingly within the distribution of all achievements. The concept presented and discussed in this paper in the context of procurement auctions takes advantage of the so-called Weighted Ordered Weighted Average (WOWA) aggregations of the partial achievements.

Keywords: multi-attribute auction, ordered weighted average, reference point method.

1 Introduction

Procurement refers to the process of obtaining goods and services required by the firm. It may be considered the acquisition of appropriate goods or services at the best possible total cost while complying with the needs of the buyer. Many types of businesses, public institutions in particular, very often define procurement processes with intention to promote fair and open competition for their business and minimize exposure to secret agreements and fraud. For this reason a competitive bidding is widely deployed in procurement. Competitive bidding is the process by which multiple suppliers submit competing offers to supply the goods or services requested by the firm, which then awards business to the supplier(s) based on these offers. The emergence of Internet-based communication between firms has enabled them to effectively organize competitive bidding events. These may be either “one shot” or “dynamic”. The latter allows several rounds of bidding thus forming the so-called auction process. An auction format is simply a set of predefined rules outlining how bids will be submitted and how

the winner and payments will be subsequently determined based on the bids [27]. Based on bids and asks placed by market participants, resource allocation and prices are determined. In electronic commerce transactions, auctions are conducted by software agents that negotiate on behalf of buyers and sellers [24,9]. The various auction protocols include English, First-price Sealed Bid, Dutch, Vickrey and others [11]. The procurement auction is a reverse auction, i.e. it is a type of auction in which the roles of buyers and sellers are reversed. In an ordinary auction (also known as a forward auction), buyers compete to obtain a good or service. Typically, as a result of this competition, the price increases during the auction. In a reverse auction, sellers compete to provide the buyer with their good or service. Typically, as a result of this type of competition, price decreases during the auction. Reverse auction is a strategy used by many purchasing and supply management organizations for spend management, as part of strategic sourcing and overall supply management activities.

One of the key challenges of current day electronic procurement systems is to enable procurement decisions overcome a limitation to a single attribute such as cost. As a result, multi-attribute procurement has gained on importance and became popular research direction. Multi-attribute auctions allow negotiations to involve multiple attributes, i.e. they overcome the limitation of single dimension of the price and expand to other attributes like e.g. quality or reputation [21]. Usually, buyer reveals her / his preferences on the good / service to be purchased. Following that sellers compete on all attributes to win the auction. Multi-attribute auctions require several key components to automate the process [4]: a preference model to let the buyer express his preferences, a multicriteria aggregation model to let the buyer agent select the best offer, a decision making component to let the buyer agent formulate her / his asks. Buyer's preferences are expressed by defining a set of relevant attributes, the domain of each attribute, and criteria which are evaluation functions that allocate a score for every possible values of a relevant attribute. Most multicriteria aggregation models used in multi-attribute negotiations are scoring functions based on a weighted sum [5,6,13,22]. It is well-known, however, that the weighted sum, which is the simplest multicriteria aggregation model, suffers from several drawbacks. This is essentially due to the fact that the weighted sum is a totally compensatory aggregation model with trade-off weights which are difficult to obtain and to interpret in the case of more than two attributes [26]. In our context, a very bad value on a criterion can be compensated by a series of good values on other criteria. Such a bid could obtain a weighted sum similar to a bid with rather good scores on all criteria, while in many cases, the latter would be preferred. Moreover, the selections are unstable in the sense slight variations on the weights may change dramatically the choice of the best bid. Finally, it can be shown that some of the non-dominated solutions, called non-supported, cannot be obtained as the best proposal using the weighted sum for any possible choice of weights. This is a very severe drawback since these non-supported solutions, whose potential interest is the same as the other non-dominated solutions, are rejected only for technical reasons. In order to address these shortcomings, Bellosta et al. [3] proposed the

use of an alternative multicriteria model for the buyer's preferences, based on the Reference Point Method [25]. There was proposed a complete reverse auction mechanism based on this model where buyer's asks specify the values required on the attributes of the item at each step of the auction process. This mechanism provides more control to the buyer agent over the bidding process than with the weighted sum model. In this approach, preference information and relative importance of criteria is not expressed in terms of weights, but more directly in terms of required values on the criteria. Moreover, while in the weighted sum, any non-dominated solution can be obtained as the best proposal.

This paper is organized as follows. Section 2 recalls basic concepts from Multi-criteria Decision Analysis (MCDA). Section 3 analyses related work [3] introducing to the Reference Point Method (RPM) based multi-attribute auction mechanism where both the preference model and the multicriteria aggregation model used by the buyer agent are based on the RPM. Section 4 presents Weighted Ordered Weighted Averaging (WOWA) extension of the RPM model thus allowing to introduce the importance weighting of attributes into the buyer preference model and the corresponding procedure for the best offer selection. This section is followed by a concluding section.

2 MCDA Concepts in Procurement Auctioning Context

A single step (round) in multi-attribute procurement auction corresponds to a special case of multicriteria problem. There is a known set \mathcal{D} of A offers \mathbf{x} ($\mathbf{x} \in \mathcal{D}$). Every offer is characterized by a set of m attributes undergoing evaluation. Therefore, every offer \mathbf{x}^a has a corresponding, vector of evaluations $\mathbf{y}^a = y_1^a, \dots, y_l^a$ where y_i^a denotes evaluation corresponding to i -th attribute for a -th offer. This may be presented as in Tab. 1.

The buyer aims to select such an offer for which its corresponding evaluation is better than evaluations of other offers. In case when the evaluation space is multi-attribute ($m > 1$) judging which of the two given offers is better than the other is usually not straight-forward. In general, preferences revealed by the buyer may be expressed as a two argument relation (preference relation). This relation is a certain set of pairs of evaluations, for which one can say "this one is better than the other".

Let us take a sample evaluation $\bar{\mathbf{y}}$. If $\bar{\mathbf{y}}$ is better than all other evaluations then $\bar{\mathbf{y}}$ is called the greatest and the corresponding offer is the best. If there is no such an evaluation that it is better than $\bar{\mathbf{y}}$ then $\bar{\mathbf{y}}$ is called maximal. If $\bar{\mathbf{y}}$ is not better than any other evaluation then $\bar{\mathbf{y}}$ is called minimal. If $\bar{\mathbf{y}}$ is worse than all other evaluations then it is called the least and the corresponding offer is the worse.

The maximal element is called non-dominated and the greatest element is called dominating. Every minimal element is dominated by some element(s) and the least element is dominated by all other elements. In such case the preference relation is called a (strong) dominance relation if there exists the greatest element with respect to this relation. Relation is called a (weak) dominance relation if there is a maximal element with respect to this relation.

Table 1. Partial evaluations

Offers	Evaluations	Attributes					
		f_1	f_2	\dots	f_i	\dots	f_m
		Partial evaluations					
\mathbf{x}^1	\mathbf{y}^1	y_1^1	y_2^1	\dots	y_i^1	\dots	y_m^1
\mathbf{x}^2	\mathbf{y}^2	y_1^2	y_2^2	\dots	y_i^2	\dots	y_m^2
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
\mathbf{x}^a	\mathbf{y}^a	y_1^a	y_2^a	\dots	y_i^a	\dots	y_m^a
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
\mathbf{x}^A	\mathbf{y}^A	y_1^A	y_2^A	\dots	y_i^A	\dots	y_m^A

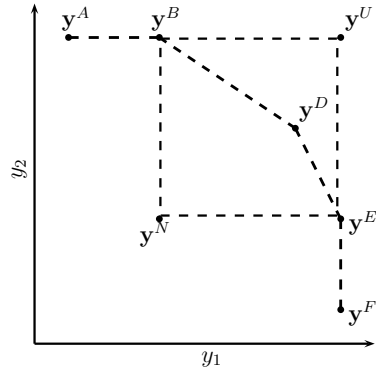


Fig. 1. Basic concepts in MCDA

An offer is called an Optimal Solution (OS) if the corresponding evaluation is dominating in a sense of a defined preference relation. Referring to Fig. 1 one can see that offer \mathbf{x}^U could be an OS as well as all potential offers that could be placed, say, north-east of \mathbf{x}^U . However, they are not feasible, there was no seller bidding with such an offer, or even such a combination of evaluations of attributes is not possible. Efficient Solution (ES) is such an offer for which the corresponding evaluation is non-dominated (maximal element in preference relation). According to the preference relation offer \mathbf{x}^0 is an ES when there is no other offer in the available such that it is not worse for every attribute and it is better at least for one attribute. If the problem has many non-dominated evaluations (what means many ESs) one could draw a conclusion that model of preferences needs improvement. Big number of ESs does not really support the buyer in choosing the offerent.

Fig. 1 illustrates basic concepts in MCDA. Axes y_1 and y_2 span the space of a two attribute problem. Points $\mathbf{y}^A, \mathbf{y}^B, \mathbf{y}^D, \mathbf{y}^E, \mathbf{y}^F$ correspond to evaluation of five arbitrary offers. From these, only evaluations $\mathbf{y}^B, \mathbf{y}^D$ and \mathbf{y}^E correspond to ESs. Obviously, offer evaluated with \mathbf{y}^A is worse than offer corresponding to \mathbf{y}^B and offer given \mathbf{y}^F is worse than offer scored with \mathbf{y}^E . Evaluation \mathbf{y}^U (that is called the utopia) is a virtual point that is constructed by using best value for each attribute separately. Similarly, evaluation \mathbf{y}^N (that is called the nadir) is constructed by taking the worst value for each attribute (from the ESs set only).

3 The RPM Based Auction Model

In order to specify the procedure used to select the best bid in detail, one needs to assume some multicriteria solution concept well adjusted to the buyer’s preferences. This can be achieved with the so-called quasi-satisficing approach proposed and developed mainly by Wierzbicki [25] as the Reference Point Method (RPM). This approach has led to efficient implementations with many successful applications in various domains [12,26]. The RPM is an interactive technique

that can be described on the ground of auctioning as follows. The buyer specifies requirements in terms of reference levels for all attributes, i.e., she / he introduces reference values for several individual evaluations. Depending on the specified reference levels, a special Scalarizing Achievement Function (SAF) is built. The SAF may be directly interpreted as expressing utility to be maximized. Maximization of the SAF generates an ES to the multicriteria problem. The computed ES is presented to the buyer as the current solution in a form that allows comparison with the previous ones and modification of the reference levels if necessary.

The SAF can be viewed as two-stage transformation of the original evaluations. First, the strictly monotonic Partial Achievement Functions (PAFs) are built to measure individual performance on each attribute with respect to given reference levels. Having all the evaluations transformed into a uniform scale of individual achievements they are aggregated at the second stage to form a unique scalarization. The RPM is based on the so-called augmented (or regularized) max-min aggregation. Thus, the worst individual achievement is essentially maximized but the optimization process is additionally regularized with the average achievement. The generic SAF takes the following form [25]:

$$S(\mathbf{y}) = \min_{1 \leq i \leq m} \{s_i(y_i)\} + \frac{\varepsilon}{m} \sum_{i=1}^m s_i(y_i) \tag{1}$$

where ε is an arbitrary small positive number and $s_i : R \rightarrow R$, for $i = 1, 2, \dots, m$, are the PAFs measuring actual achievement of the individual evaluations y_i with respect to the corresponding reference levels. Let a_i denote the partial achievement for the i -th evaluation ($a_i = s_i(y_i)$) and $\mathbf{a} = (a_1, a_2, \dots, a_m)$ represent the achievement vector. Various functions s_i provide a wide modeling environment for measuring partial achievements [26]. The basic RPM model is based on a single vector of the reference levels, the aspiration vector \mathbf{r}^a and the Piecewise Linear (PWL) functions s_i .

Real-life applications of the RPM methodology usually deal with more complex PAFs defined with more than one reference point [26]. In particular, the models taking advantage of two reference vectors: vector of aspiration levels \mathbf{r}^a and vector of reservation levels \mathbf{r}^r [12] are used, thus allowing the buyer to specify requirements by introducing acceptable and required values for several evaluations. The PAF s_i can be interpreted then as a measure of the buyer's satisfaction with the current value of evaluation the i -th criterion. It is a strictly increasing function of evaluation y_i with value $a_i = 1$ if $y_i = r_i^a$, and $a_i = 0$ for $y_i = r_i^r$. Various functions can be built meeting those requirements. We use the PWL PAF [18]:

$$s_i(y_i) = \begin{cases} \gamma(y_i - r_i^r)/(r_i^a - r_i^r), & y_i \leq r_i^r \\ (y_i - r_i^r)/(r_i^a - r_i^r), & r_i^r < y_i < r_i^a \\ \alpha(y_i - r_i^a)/(r_i^a - r_i^r) + 1, & y_i \geq r_i^a \end{cases} \tag{2}$$

where α and γ are arbitrarily defined parameters satisfying $0 < \alpha < 1 < \gamma$.

At each round of a reverse auction, the buyer agent collects all the bids, selects the best one as the reference bid for the next round and formulates the counterproposal. The definition of counterproposals is based on the beat-the-quote rule which specifies that any new bid must beat the best bid received at the previous round. In the standard case of one-dimensional (price) auction, this rule can simply be implemented by communicating to the sellers the evaluation of the best current bid augmented by a minimal increment Δ . Sellers are then asked to send new bids whose evaluation is at least as good as this augmented evaluation. The same may be applied to a scalar aggregation function but this would require that sellers know and implement the buyer's evaluation model. As shown in [3] the RPM based auction mechanism satisfies the beat-the-quote rule without revealing the buyer's evaluation model to the sellers. This is achieved through the use of reservation levels set as the best bid's achievements augmented by a minimal increment Δ and communicated to the sellers as the minimal requirements.

In the multi-attribute procurement auction protocol buyer agent gathers information about buyers preferences. These include the value functions and respective aspiration and reservation levels. Additionally, the time of the closing of the auction is set. The buyer agent also specifies an increment used to define counterproposals and time span of a single negotiation round. All this information is sent to seller agents who in reply send initial proposal (or abort their participation). Repeatedly, until the auction ends, after evaluation of all proposals the best seller is marked as active and other sellers are updated with new reservation levels to allow further bidding. The auction ends with success when there is only one seller left in the competition or when the closing time of the auction is reached. The auction end with failure if there is no seller left in the competition.

4 WOWA Extension of the RPM

The crucial properties of the RPM are related to the max-min aggregation of partial achievements while the regularization is only introduced to guarantee the aggregation monotonicity. Unfortunately, the distribution of achievements may make the max-min criterion partially passive when one specific achievement is relatively very small for all the solutions. Maximization of the worst achievement may then leave all other achievements unoptimized. Nevertheless, the selection is then made according to linear aggregation of the regularization term instead of the max-min aggregation, thus destroying the preference model of the RPM [17].

In order to avoid inconsistencies caused by the regularization, the max-min solution may be regularized according to the ordered averaging rules [28]. This is mathematically formalized as follows. Within the space of achievement vectors we introduce map $\Theta = (\theta_1, \dots, \theta_m)$ which orders the coordinates of achievements vectors in a nonincreasing order, i.e., $\Theta(a_1, \dots, a_m) = (\theta_1(\mathbf{a}), \theta_2(\mathbf{a}), \dots, \theta_m(\mathbf{a}))$ iff there exists a permutation τ such that $\theta_i(\mathbf{a}) = a_{\tau(i)}$ for all i and $\theta_1(\mathbf{a}) \geq$

$\theta_2(\mathbf{a}) \geq \dots \geq \theta_m(\mathbf{a})$. The standard max-min aggregation depends on maximization of $\theta_m(\mathbf{a})$ and it ignores values of $\theta_i(\mathbf{a})$ for $i \leq m - 1$. In order to take into account all the achievement values, one needs to maximize the weighted combination of the ordered achievements thus representing the so-called Ordered Weighted Averaging (OWA) aggregation [28]. Note that the weights are then assigned to the specific positions within the ordered achievements rather than to the partial achievements themselves. With the OWA aggregation one gets the following RPM model:

$$\max \left\{ \sum_{i=1}^m w_i \theta_i(\mathbf{a}) : a_i = s_i(f_i(\mathbf{x})) \ \forall i, \ \mathbf{x} \in Q \right\} \tag{3}$$

where $w_1 < w_2 < \dots < w_m$ are positive and strictly increasing weights. Actually, they should be significantly increasing to represent regularization of the max-min order. Note that the standard RPM model with the scalarizing achievement function (1) can be expressed as the OWA model (3) with weights $w_1 = \dots = w_{m-1} = \varepsilon/m$ and $w_m = 1 + \varepsilon/m$ thus strictly increasing in the case of $m = 2$. Unfortunately, for $m > 2$ it abandons the differences in weighting of the largest achievement, the second largest one etc ($w_1 = \dots = w_{m-1} = \varepsilon/m$). The OWA RPM model (3) allows one to differentiate all the weights by introducing increasing series (e.g. geometric ones).

Typical RPM models allow weighting of several achievements only by straightforward rescaling of the achievement values. The OWA RPM model enables one to introduce importance weights to affect achievement importance by rescaling accordingly its measure within the distribution of achievements as defined in the so-called Weighted OWA (WOWA) aggregation [23]. Let $\mathbf{w} = (w_1, \dots, w_m)$ be a vector of preferential (OWA) weights and let $\mathbf{p} = (p_1, \dots, p_m)$ denote the vector of importance weights ($p_i \geq 0$ for $i = 1, 2, \dots, m$ as well as $\sum_{i=1}^m p_i = 1$). The corresponding Weighted OWA aggregation of achievements $\mathbf{a} = (a_1, \dots, a_m)$ is defined as follows:

$$A_{\mathbf{w}, \mathbf{p}}(\mathbf{a}) = \sum_{i=1}^m \omega_i \theta_i(\mathbf{a}), \quad \omega_i = w^* \left(\sum_{k \leq i} p_{\tau(k)} \right) - w^* \left(\sum_{k < i} p_{\tau(k)} \right) \tag{4}$$

where w^* is an increasing function that interpolates points $(\frac{i}{m}, \sum_{k \leq i} w_k)$ together with the point (0.0) and τ representing the ordering permutation for \mathbf{a} (i.e. $a_{\tau(i)} = \theta_i(\mathbf{a})$). Moreover, function w^* is required to be a straight line when the point can be interpolated in this way. Due to this requirement, the WOWA aggregation covers the standard weighted mean with weights p_i as a special case of equal preference weights ($w_i = 1/m$ for $i = 1, 2, \dots, m$). Function w^* can be defined by its generation function

$$g(\xi) = mw_i \quad \text{for } (i - 1)/m < \xi \leq i/m, \quad i = 1, 2, \dots, m \tag{5}$$

with the formula $w^*(\alpha) = \int_0^\alpha g(\xi) \, d\xi$.

Introducing breakpoints $\alpha_i = \sum_{k \leq i} p_{\tau(k)}$ and $\alpha_0 = 0$ allows us to express

$$\omega_i = \int_0^{\alpha_i} g(\xi) \, d\xi - \int_0^{\alpha_{i-1}} g(\xi) \, d\xi = \int_{\alpha_{i-1}}^{\alpha_i} g(\xi) \, d\xi$$

Therefore, the WOWA may be expressed with more direct formula where preferential (OWA) weights w_i are applied to averages of the corresponding portions of ordered achievements (quantile intervals) according to the distribution defined by importance weights p_i [20]:

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{a}) = \sum_{i=1}^m w_i m \int_{\frac{i-1}{m}}^{\frac{i}{m}} F_{\mathbf{a}}^{(-1)}(\xi) d\xi \tag{6}$$

where $\overline{F}_{\mathbf{a}}^{(-1)}$ is the stepwise function $\overline{F}_{\mathbf{a}}^{(-1)}(\xi) = \theta_i(\mathbf{a})$ for $\beta_{i-1} < \xi \leq \beta_i$. It can also be mathematically formalized as follows. First, we introduce the right-continuous cumulative distribution function (cdf) $F_{\mathbf{a}}(d) = \sum_{i=1}^m p_i \delta_i(d)$ where $\delta_i(d) = 1$ if $a_i \leq d$ and 0 otherwise. Next, we introduce the quantile function $F_{\mathbf{a}}^{(-1)} = \inf \{\eta : F_{\mathbf{a}}(\eta) \geq \xi\}$ for $0 < \xi \leq 1$ as the left-continuous inverse of $F_{\mathbf{a}}$, ie., $F_{\mathbf{a}}^{(-1)}(\xi) = \inf \{\eta : F_{\mathbf{a}}(\eta) \geq \xi\}$ for $0 < \xi \leq 1$, and finally $\overline{F}_{\mathbf{a}}^{(-1)}(\xi) = F_{\mathbf{a}}^{(-1)}(1-\xi)$.

Formula (6) defines the WOWA value applying preferential weights w_i to importance weighted averages within quantile intervals. It may be reformulated to use the tail averages which are LP computable. Indeed, one may get the following model [17] for the WOWA RPM with PWL PAFs (2):

$$\begin{aligned} \max \sum_{k=1}^m w'_k z_k \quad \text{s.t.} \quad & z_k = kt_k - m \sum_{i=1}^m p_i d_{ik} & \forall k \\ & \mathbf{x} \in Q, y_i = f_i(\mathbf{x}) & \forall i \\ & a_i \geq t_k - d_{ik}, d_{ik} \geq 0 & \forall i, k \\ & a_i \leq \gamma(y_i - r_i^r)/(r_i^a - r_i^r) & \forall i \\ & a_i \leq (y_i - r_i^r)/(r_i^a - r_i^r) & \forall i \\ & a_i \leq \alpha(y_i - r_i^a)/(r_i^a - r_i^r) + 1 & \forall i \end{aligned} \tag{7}$$

thus allowing for implementation of the entire WOWA RPM model as an LP expansion of the original problem. Although while using the WOWA RPM aggregation to find out the best bid we are dealing with a finite discrete set of bids and the direct WOWA formula (4) with predefined piecewise linear function w^* can be effectively used.

The WOWA aggregation with positive weights is strictly increasing [17]. Therefore, similar to the standard RPM based auction mechanism [3], the WOWA RPM based mechanism also satisfies the beat-the-quote rule without revealing the buyer’s evaluation model to the sellers. This is achieved through the use of reservation levels set as the best bid’s achievements augmented by a minimal increment Δ and communicated to the sellers as the minimal requirements. Note that revealing the importance weights still does not reveal completely the buyer’s preference model. Thus one may make the attributes importance weights commonly available to meet possible requirement for some public sector procurement auctions.

5 Conclusions

This paper describes a multi-attribute auction mechanism based on reference points with the non-compensatory importance weights for several attributes (criteria). As with the weighted sum model the buyer's preferences include value functions. However, compensatory weights associated with attributes are replaced by aspiration levels that represent the required values on the attributes of the item to be purchased and the non-compensatory importance weights associated with attributes. Similar to the unweighted RPM multi-attribute auction mechanism [3], auctions are conducted using reservation levels that express the minimum values acceptable on the attributes. Since the WOVA aggregation with positive weights is strictly monotonic, this way of defining counterproposal ensures a successive refinement of the best bids in each round and thereby preserves the efficiency of the RPM auction. Thus the mechanism addresses the shortcomings of the weighted sum model while allowing to take into account the importance weighting of several attributes.

Acknowledgment. The research was partially supported by the Polish National Budget Funds 2010–2013 for science under the grant N N514 044438.

References

1. Bapna, R., Goes, P., Gupta, A.: A theoretical and empirical investigation of multi-item on-line auctions. *Information Technology and Management* 1, 1–23 (2000)
2. Bapna, R., Goes, P., Gupta, A.: Insights and analyses of on-line auctions. *Communications ACM* 44, 43–50 (2001)
3. Bellosta, M.-J., Brigui, I., Kornman, S., Vanderpooten, D.: A multi-criteria model for electronic auctions. In: *ACM Symposium on Applied Computing (SAC 2004)*, pp. 759–765 (2004)
4. Bellosta, M.-J., Kornman, S., Vanderpooten, D.: An Agent-Based Mechanism for Autonomous Multiple Criteria Auctions. In: *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2006)*, pp. 587–594 (2006)
5. Bichler, M.: An experimental analysis of multi-attribute auctions. *Decision Support Systems* 29, 249–268 (2000)
6. Bichler, M., Kalagnanam, J.: Configurable offers and winner determination in multi-attribute auctions. *Eur. J. Opnl. Res.* 160, 380–394 (2005)
7. Chandrashekar, T.S., Narahari, Y., Rosa, C.H., Kulkarni, D.M., Tew, J.D., Dayama, P.: Auction based mechanisms for electronic procurement. *IEEE Trans. Automation Sci.* 4, 297–321 (2006)
8. Hochner, G., Bichler, M., Davenport, A., Kalagnanam, J.: Industrial procurement auctions. In: Cramton, P., Shoham, Y., Steinberg, R. (eds.) *Combinatorial Auctions*, pp. 593–612. The MIT Press, Cambridge (2006)
9. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., Wooldridge, M.: Automated negotiation: prospects, method and challenges. *Int. J. Group Decision and Negotiation* 10, 199–215 (2001)

10. Kameshwaran, S., Narahari, Y., Rosa, C.H., Kulkarni, D.M., Tew, J.D.: Multiattribute electronic procurement using goal programming. *Eur. J. Opnl. Res.* 179, 518–536 (2007)
11. Krishna, V.: *Auction Theory*. Academic Press, San Francisco (2002)
12. Lewandowski, A., Wierzbicki, A.P.: *Aspiration Based Decision Support Systems – Theory, Software and Applications*. Springer, Berlin (1989)
13. Morris, J., Maes, P.: Sardine: An agent-facilitated airline ticket bidding system. 4th International Conference on Autonomous Agents (Agents 2000), Barcelona, Spain (2000)
14. Narahari, Y., Garg, D., Narayanam, R., Prakash, H.: *Game Theoretic Problems in Network Economics and Mechanism Design Solutions*. Springer, Heidelberg (2009)
15. Nisam, N., Roughgarden, T., Tardos, E., Vazirani, V.V. (eds.): *Algorithmic Game Theory*. Cambridge University Press, Cambridge (2007)
16. Ogryczak, W.: Preemptive reference point method. In: Climaco, J. (ed.) *Multicriteria Analysis — Proceedings of the XIth International Conference on MCDM*, pp. 156–167. Springer, Berlin (1997)
17. Ogryczak, W., Kozłowski, B.: Reference Point Method with Importance Weighted Ordered Partial Achievements. *TOP(2009)* (forthcoming), doi: 10.1007/s11750-009-0121-4
18. Ogryczak, W., Studziński, K., Zorychta, K.: DINAS: A Computer-Assisted Analysis System for Multiobjective Transshipment Problems with Facility Location. *Comp. Opns. Res.* 19, 637–647 (1992)
19. Ogryczak, W., Śliwiński, T.: On solving linear programs with the ordered weighted averaging objective. *Eur. J. Opnl. Res.* 148, 80–91 (2003)
20. Ogryczak, W., Śliwiński, T.: On Optimization of the Importance Weighted OWA Aggregation of Multiple Criteria. In: Gervasi, O., Gavrilova, M.L. (eds.) *ICCSA 2007, Part I. LNCS*, vol. 4705, pp. 804–817. Springer, Heidelberg (2007)
21. Petric, A., Jezic, G.: Reputation Tracking Procurement Auctions. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) *ICCCI 2009. LNCS*, vol. 5796, pp. 825–837. Springer, Heidelberg (2009)
22. Teich, J.E., Wallenius, H., Wallenius, J., Zaitsev, A.: A multi-attribute e-auction mechanism for procurement: theoretical foundations. *Eur. J. Opnl. Res.* 175, 90–100 (2006)
23. Torra, V.: The weighted OWA operator. *Int. J. Intell. Syst.* 12, 153–166 (1997)
24. Torra, V., Narukawa, Y.: *Modeling Decisions Information Fusion and Aggregation Operators*. Springer, Berlin (2007)
25. Wierzbicki, A.P.: A Mathematical Basis for Satisficing Decision Making. *Math. Modelling* 3, 391–405 (1982)
26. Wierzbicki, A.P., Makowski, M., Wessels, J. (eds.): *Model Based Decision Support Methodology with Environmental Applications*. Kluwer, Dordrecht (2000)
27. Wurman, P.R., Wellman, M.P., Walsh, W.E.: Specifying rules for electronic auctions. *AI Magazine* 23, 15–23 (2002)
28. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. Systems, Man Cyber.* 18, 183–190 (1988)

ASPARAGUS - A System for Automatic SPARQL Query Results Aggregation Using Semantics

Agnieszka Ławrynowicz, Jędrzej Potoniec, Łukasz Konieczny, Michał Madziar, Aleksandra Nowak, and Krzysztof T. Pawlak

Institute of Computing Science, Poznan University of Technology, ul. Piotrowo 2, 60-965 Poznan, Poland

Abstract. We present a prototype system, named ASPARAGUS, that performs aggregation of SPARQL query results on a semantic baseline, that is by an exploitation of the background ontology expressing the semantics of the returned results. The system implements the recent research results on semantic grouping, and semantic clustering. In the former case, results are deductively grouped taking into account the subsumption hierarchy deduced by the knowledge base. In the latter case, the results are clustered that is inductively formed, based on the similarity of the individual resources. We discuss the architecture of the implemented system, its underlying technologies, and applied technical solutions.

Keywords: Semantic Web, SPARQL, ontologies, data mining.

1 Introduction

Query answering on the Web can return a large number of results that can be hardly manageable by users. Moreover, typically, only a small part of the result set is relevant to the user thus making necessary the analysis of the retrieved results to identify those relevant ones. This phenomenon is known as *information overload*. To liberate users of a tedious manual analysis job, various services have been set up, such as predefined category structures, and fully automatic search engines. The problem with predefined category structures, frequently maintained manually, is that they may easily become obsolete due to fast changes of the Web content, additionally they often cover only a small part of the existing Web sites. Search engines, a primary choice for searching the Web for the most users, may be less effective in case of broad or ambiguous queries where the results on different subtopics happen to be mixed in the one results list. A promising solution, adopted in this work, is to marry the benefits of the two mentioned approaches and generate dynamic groupings/clustering over retrieved query results.

This paper presents a prototype system based on the recent research results on the topic of *semantic aggregation* of Web query results. In [7] a new method for grouping query results was introduced, coined *semantic grouping*. Given on input a conjunctive query, and a background OWL [13] ontology, the method returns a dynamic, *runtime* categorization over ranked query results. During generation of the categorization, the method exploits the *semantics* of knowledge bases of reference (in the form of ontologies) by an application of deductive reasoning. In particular, given a grouping criterion expressed as a (complex) concept from a knowledge base of reference, results are grouped according to (part of) the subsumption hierarchy deductively obtained by considering the specified concept and the given ontology. In this way, the results can be shown and navigated similarly to a faceted search.

In [10], an idea and a method were proposed for *semantic clustering* of query results, that consists in an application of data mining methods to inductively build clusters of similar results. In [11], to support such type of the results aggregation, a possible extension to SPARQL [15], a standard Semantic Web query language, was discussed. Importantly, in the 'semantic clustering' approach, the semantics of clustered objects (expressed in the background ontologies) is exploited to compute similarities or distances between the individual resources.

We have implemented the mechanisms of semantic aggregation of the query results, both deductive ('semantic grouping'), and inductive ('semantic clustering'), in the system named ASPARAGUS (*Automatic SPARQL query results Aggregation Using Semantics*), which is described in this paper.

2 A Concept of Semantic Aggregation

Let us illustrate the concepts of semantic grouping and semantic clustering of query results by means of simple examples. Assume that a user is submitting a query for weekend break offers, and would like to have the results grouped by a destination criterion. Classically, aggregation abilities are provided by SQL-like GROUP BY clause, which is to be supported in a new version of SPARQL. A hypothetical SPARQL query representing the abovementioned user information needs could have the following form:

```
Q1 = SELECT ?x ?y WHERE { ?x rdf:type :WeekendBreakOffer .
?x :hasDestination ?y } GROUP BY ?y
```

However, the classical GROUP BY semantics, which is to partition the results by identical values, is not always suitable. In the case, when destination is represented by a town name, one group for each town name would be created by GROUP BY (as shown in Fig. 1 a)), which could result in too many groups to be easily managed by the user. In turn, if the destination criterion would be represented by an ontology concept *Destination*, being a root of a concept hierarchy which we would like to exploit for grouping the results, the towns would be annotated by its subconcepts such as *EuropeanDestination*, *PolishDestination*, *SeasideDestination*, *SkiResort* etc., then it would be possible to aggregate

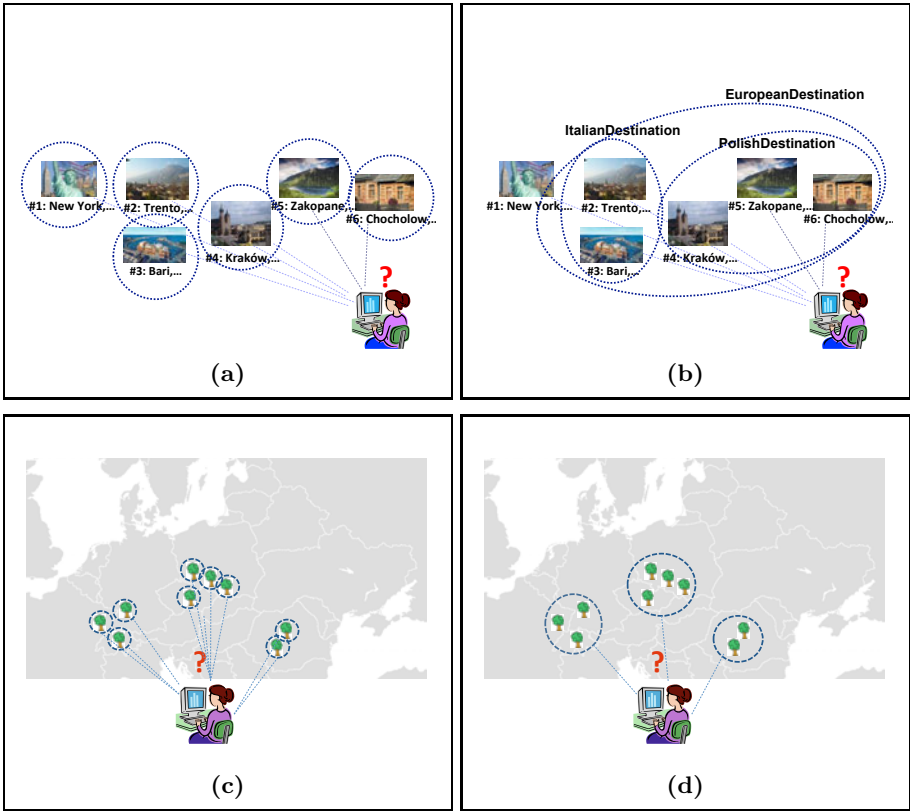


Fig. 1. A concept of semantic, and non-crisp aggregation: (a) classical grouping vs (b) semantic grouping, and (c) classical grouping vs (d) clustering

the results into a smaller number of 'semantic' groups e.g. into Polish or Italian destinations or into seaside or ski resort destinations. Moreover, since such concepts would frequently form a hierarchy (e.g. PolishDestination would be a subconcept of EuropeanDestination) it would be also possible to exploit such taxonomic relationships during semantic grouping (see Fig. 1 b)). In [7] a new clause CATEGORIZE BY has been introduced to express the semantics of such grouping:

```
Q2 = SELECT ?x ?y WHERE { ?x rdf:type :WeekendBreakOffer .
?x :hasDestination ?y } CATEGORIZE BY ?y
```

Also semantic clustering may exploit ontologies during grouping the results, but instead of purely deductive services, it exploits also semantic similarity measures to inductively build clusters of similar individuals. Let us assume this time, that the user searches for natural monuments which (s)he would like to have grouped

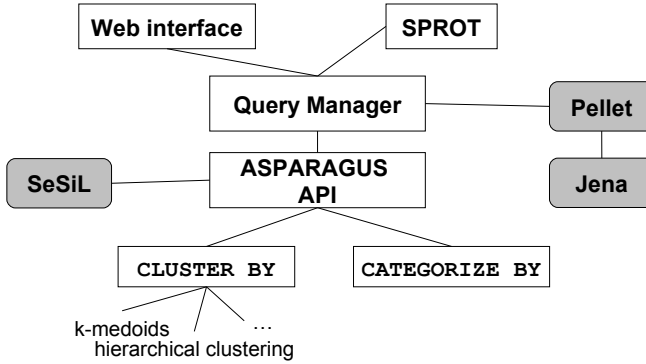


Fig. 2. The architecture of ASPARAGUS

by their mutual proximity. Would be then grouping by longitude and latitude what the user actually expected? Since no two monuments share the same coordinates, one group for each monument would again be created (Fig. 1 c). By relaxing strict, boolean criteria of the GROUP BY by means of similarity measures, we may extend the framework of 'semantic aggregation' to encompass inductive, clustering abilities. For this purpose, an extension of SPARQL by CLUSTER BY clause has been proposed [11] to be able to issue queries in the spirit of the one below (reflected also in Fig. 1 d):

```

Q3 = SELECT ?x ?latitude ?longitude WHERE ?x rdf:type :NaturalMonument . ?x
geo:lat ?latitude . ?x geo:long ?longitude . CLUSTER BY ?latitude ?longitude
    
```

3 Description of the System

An overview of the architecture of ASPARAGUS is presented in Fig. 2. Central to the system is *Query Manager*, responsible for task scheduling, and invoking of subsequent operations. To manipulate, and reason with knowledge bases, and to answer SPARQL queries, a reasoner or semantic repository is needed. Currently ASPARAGUS uses Pellet reasoner¹ [16] and Jena API². The implemented grouping algorithms (CATEGORIZE BY and CLUSTER BY modes) are linked by a common programming interface of ASPARAGUS. The interface mediates also between the system and the SeSiL library that implements a set of semantic similarity measures, exploited during clustering query results. An interaction with the user may be performed in two ways: i) by a Web interface, and ii) (semi-)automatically by use of the standard SPROT interface for submitting SPARQL queries. Subsequently we describe the components of the system in more detail.

¹ <http://clarkparsia.com/pellet/>

² <http://jena.sourceforge.net/>

3.1 Query Manager

Query Manager schedules execution of queries submitted either through WWW interface or SPARQL endpoint. A *Query* is an object that is delivered to *Query Manager* in the form containing: a) the query full text (consisting of standard SPARQL query part, the mode of aggregation that is **CATEGORIZE BY** or **CLUSTER BY**, the list of variables to be used during aggregation, and an algorithm to be used with its parameters), b) an indication of a SPARQL endpoint to retrieve data from, and c) list of URLs or access paths of files to be read in to a default graph.

Query processing consist in performing the following steps: 1) Creation of Pellet's helper objects, that contain SPARQL query, and a list of URLs to retrieve data from, 2) Generation of the query execution plan, 3) Query execution and retrieving the results, 4) Aggregation of the results with use of a previously indicated algorithm, 5) Marking the query as finished. In case of an error during execution of any of the above steps, it is recorded as an answer, and the query is marked as finished.

Since ASPARAGUS works within a J2EE application server, *Query Manager* is run as a stand-alone thread that uses a multi-threaded queue. The source code, responsible for preparing a query, invokes an appropriate method of a shared *Query Manager* object, that puts it to the queue, and waits for a signal on the availability of the query answer in the query object. Thanks to the use of the multi-threaded queue, when there are no tasks to process, the processing thread stays in the sleep state. Such construction optimizes the use of system resources, since there is no risk of executing more queries than beforehand envisaged (dependently of the number of simultaneously used *Query Manager* objects), and it does not waste the resources during waiting for the new task.

3.2 CATEGORIZE BY

CATEGORIZE BY clause extends the standard SPARQL syntax by an ability to group the results along a hierarchy of classes (concepts) of which particular query result instances are members [7]. The syntax of the **CATEGORIZE BY** clause (in *EBNF*) is:

```
query ::= select query, categorize by
categorize by ::= "CATEGORIZE BY", variables
variables ::= variable, {" ", variable}
```

where `select query` represents valid **SELECT** sentence, and `variable` any variable appearing also in **SELECT** statement (possibly also via the use of symbol `*`).

3.3 CLUSTER BY

CLUSTER BY clause extends the standard SPARQL syntax by an ability to cluster the results by means of data mining algorithms working on-line. The syntax of **CLUSTER BY** clause (in *EBNF*) is similar to the syntax of **CATEGORIZE BY**:

```

query ::= select query, cluster by
cluster by ::= "CLUSTER BY", variables, [using]
variables ::= variable, {" ", variable}
using ::= "USING <", uri, ">"
uri ::= "http://semantic.cs.put.poznan.pl/asparagus/cluster-by/hierarchical" |
        "http://semantic.cs.put.poznan.pl/asparagus/cluster-by/kmedoids"

```

Currently ASPARAGUS implements two clustering algorithms, a classical Agglomerative Hierarchical Clustering type algorithm, extended by an ability to produce semantic descriptions of clusters, and a variant of k-medoids algorithm proposed in [8], extended with the possibility to apply similarity measures for datatypes (please note, that CATEGORIZE BY groups objects exclusively based on their membership to classes described in the background knowledge base). Invoking an appropriate algorithm is steered by an appropriate choice in the USING clause. In order to measure similarity between clustered individuals, SESIL library is used. SESIL (*Semantic Similarity Measures Library*) has been developed by the Poznan University of Technology since 2010 within EU FP7 project e-LICO, and to this end, it implements a set of (semantic) kernel functions (e.g. the ones proposed in [9,4]) to compute the similarity between individual resources described by ontologies represented in description logics [1].

3.4 Interface

Web interface. The Web interface of ASPARAGUS has been implemented using *Google Web Toolkit (GWT)* framework. It is available at <http://semantic.cs.put.poznan.pl/Asparagus/>. The interface (see Fig. 3) 1) enables entering a text of SPARQL query, 2) displays a dynamically generated, navigable hierarchy on top of the retrieved results, 3) displays the content of a group the user navigated to, and 4) enables the specification of a set of options such as: indication of a SPARQL endpoint or HTTP addresses to retrieve data from, indication of a local file to upload temporarily to the system, displaying or hiding empty groups, help, hints and examples. Fig. 3 presents screenshots of the Web interface of ASPARAGUS.

The screenshots present the Web interface displaying the results of an invocation of the three currently available modes of aggregation: by means of CATEGORIZE BY (Fig. 3a), by means of agglomerative hierarchical clustering (AHC) (Fig. 3b), and k-medoids based clustering (Fig. 3c). The examples use data extracted from part of DBpedia [3] dataset. As can be seen from the figures, CATEGORIZE BY mode groups the data in full accordance with the DBpedia ontology. The results of AHC algorithm clearly reflect its way of working: it builds a hierarchy of clusters from the individual elements by progressively merging clusters. It is reflected on the Fig. 3b), where the groups aggregating individual elements of type *BodyOfWater* or *Park* are first merged into one group, which is further merged with the group aggregating individuals of type *Building*. In case of k-medoids type algorithm, a medoid of each group (a most representative element in the group) is used as its description.

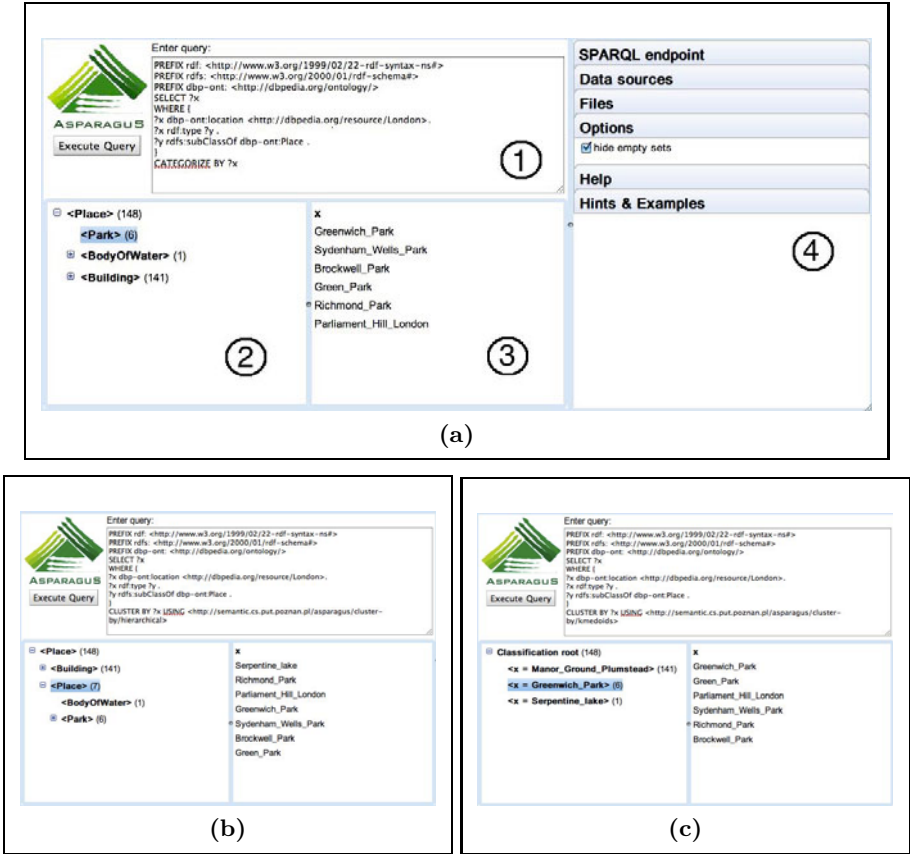


Fig. 3. Screenshots of the Web interface of ASPARAGUS

SPARQL endpoint. SPARQL endpoint of ASPARAGUS is available at <http://semantic.cs.put.poznan.pl/Asparagus/sparql/>. It offers an access to the same algorithms and methods which are also made available by the Web interface. Queries should be submitted in accordance to the SPARQL protocol. The SPARQL endpoint handles HTTP requests, while the protocol SOAP is not handled. A query may be submitted through GET or POST method in parameter query. The query must be in application/x-www-form-urlencoded format. With default-graph-uri parameter, used an arbitrary number of times, it is possible to indicate sources to be loaded into a default graph (it is equivalent to using the FROM clause in the query. Parameter named-graph-uri is not handled, but to obtain its functionality FROM NAMED clause may be used in the query. The result generated by the ASPARAGUS SPARQL endpoint is compatible with SPARQL query results XML format [2].

ASPARAGUS is licensed under the terms of GNU AGPL licence.

4 Related Work

The work related to ours falls in the following topics: Web search results clustering systems, and user interfaces for SPARQL querying.

Web search results clustering systems [5] group the results returned by a search engine into a hierarchy of labeled clusters gathering the items on the same topic. Classically, such systems have been designed to operate on unstructured, textual data of Web documents (e.g. snippets). In turn, our system has been designed to operate on structured, RDF query results. Similarly as in our case, the problem of generating meaningful labels to constructed clusters is important in the context of Web clustering engines (see for example Carrot2³ [14], and Vivisimo Clusty/Yippy⁴), but in the latter case this is achieved via NLP techniques, and not by use of background ontologies.

Since querying an RDF knowledge base with SPARQL queries is generally not considered as an end user task, various query interfaces have been proposed, most often dedicated to a specific knowledge base, and of the form of Web forms. The advantage of such interfaces is that they hide the complexity of underlying knowledge bases from the user. However, this comes with a price of loosing flexibility: they may be vulnerable to schema changes, they may enable only selected types of queries, and hence do not allow for broader exploration of underlying RDF graphs.

More flexible, non-dedicated methods, include visual SPARQL query builders (such as SPARQL Views⁵ [6] or Virtuoso Interactive Query Builder⁶), and faceted browsing (see for example facet service of Virtuoso⁷ and OntoWiki facets⁸). Visual SPARQL query builders are aimed to help the users in creating queries with less difficulty as it would be required for writing the query text. However, this still requires some knowledge of SPARQL syntax and semantics, as well as of an underlying schema. Therefore the builders are more appropriate for developers than for end users. Faceted browser interfaces are flexible enough to be used on top of arbitrary SPARQL endpoints, but in turn the types of queries that may be issued through them by the users are limited. In particular, they may not be sufficient to express very complex queries that are not expressible by just setting a combination of filters to the facets, for example a query expressing a restriction on a resource related to the main resource of interest (e.g. "Weekend break offers that are located in a town that has a museum of art"), or a restriction involving a value not frequently occurring in queries, and thus not available in a given facet space.

ASPARAGUS may be used by users knowledgeable of SPARQL syntax directly to express complex queries and get easier insight to the results. The users

³ <http://www.carrot2.org>

⁴ <http://search.yippy.com>

⁵ http://drupal.org/project/sparql_views

⁶ <http://wikis.openlinksw.com/dataspace/owiki/wiki/OATWikiWeb/> InteractiveSPARQLQueryBuilder

⁷ <http://dbpedia.org/fct/>

⁸ <http://ontowiki.net>

having less knowledge about underlying schema may want to submit a CLUSTER BY query to let the system automatically find the groupings of the results. The users aware of the schema of the queried datasets may prefer to submit a CATEGORIZE BY query, with a similar goal as such of submitting an SQL GROUP BY query - to get a desirable aggregation of the results. However, a design and implementation of ASPARAGUS make it also able to be used as a component of a more complex system, where it would be used to answer queries issued through a visual user interface dedicated to a particular application and/or a knowledge base, where SPARQL syntax is hidden from the user.

5 Conclusions and Future Work

In this paper we described a first release of a prototype system called ASPARAGUS that implements novel algorithms that involve deductive and inductive reasoning to semantically aggregate the results of SPARQL queries.

Importantly, in the design of the system, we decided to exploit the peculiar feature of the Semantic Web datasets – possible availability of the background ontologies expressing semantics of the data. The resulting system that is built is capable to leverage on this availability while processing the query results. The system provides a framework for testing various configurations of semantic aggregation solution components, such as similarity measures, categorization, and clustering algorithms etc. on arbitrary Semantic Web datasets enriched with ontologies.

Besides that it is released with its own Web user interface, ASPARAGUS may be also integrated as a part of a more complex system, e.g. as a middleware between SPARQL query engines/endpoints and a customized user interface, specific to some knowledge base, what we plan in the future work. The plans for future extensions of ASPARAGUS include making it compatible with further reasoners and semantic repositories, as well as various extensions of generating groupings and their labels, e.g. refinement operator based ones proposed in [12], or the ones assuming an individual (nominal class) as the label of a group. In the nearest future we will also work on improving the data acquisition capabilities of the system. This will include an investigation into suitable solutions for retrieving relevant parts of RDF datasets from SPARQL endpoints for meaningful aggregation of the query results.

Acknowledgements. This work is partially supported by Polish Ministry of Science and Higher Education (grant number N N516 186437) and by European Community 7th framework program ICT-2007.4.4 (grant number 231519 "e-LICO: An e-Laboratory for Interdisciplinary Collaborative Research in Data Mining and Data-Intensive Science").

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook. Cambridge University Press, Cambridge (2003)

2. Beckett, D., Broekstra, J.: SPARQL Query Results XML Format. W3C Recommendation (2008), <http://www.w3.org/TR/rdf-sparql-XMLres/>
3. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the Web of Data. *J. Web Sem.* 7(3), 154–165 (2009)
4. Bloehdorn, S., Sure, Y.: Kernel methods for mining instance data in ontologies. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) *ASWC 2007 and ISWC 2007*. LNCS, vol. 4825, pp. 58–71. Springer, Heidelberg (2007)
5. Carpineto, C., Osinski, S., Romano, G., Weiss, D.: A survey of Web clustering engines. *ACM Comput. Surv.* 41(3) (2009)
6. Clark, L.: SPARQL Views: A visual SPARQL query builder for Drupal. In: 9th International Semantic Web Conference, ISWC 2010 (November 2010), <http://data.semanticweb.org/conference/iswc/2010/paper/518>
7. d’Amato, C., Fanizzi, N., Lawrynowicz, A.: Categorize by: Deductive aggregation of semantic web query results. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) *ESWC 2010*. LNCS, vol. 6088, pp. 91–105. Springer, Heidelberg (2010)
8. Fanizzi, N., d’Amato, C., Esposito, F.: Conceptual clustering and its application to concept drift and novelty detection. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) *ESWC 2008*. LNCS, vol. 5021, pp. 318–332. Springer, Heidelberg (2008)
9. Fanizzi, N., d’Amato, C.: A declarative kernel for \mathcal{ALC} concept descriptions. In: Esposito, F., Raś, Z.W., Malerba, D., Semeraro, G. (eds.) *ISMIS 2006*. LNCS (LNAI), vol. 4203, pp. 322–331. Springer, Heidelberg (2006)
10. Lawrynowicz, A.: Grouping results of queries to ontological knowledge bases by conceptual clustering. In: Nguyen, N.T., Kowalczyk, R., Chen, S.M. (eds.) *ICCCI 2009*. LNCS, vol. 5796, pp. 504–515. Springer, Heidelberg (2009)
11. Lawrynowicz, A.: Query results clustering by extending SPARQL with CLUSTER BY. In: Meersman, R., Herrero, P., Dillon, T.S. (eds.) *OTM 2009 Workshops*. LNCS, vol. 5872, pp. 826–835. Springer, Heidelberg (2009)
12. Lawrynowicz, A., d’Amato, C., Fanizzi, N.: A refinement operator based method for semantic grouping of conjunctive query results. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) *KES 2010*. LNCS, vol. 6278, pp. 359–368. Springer, Heidelberg (2010)
13. McGuinness, D.L., van Harmelen, F.: OWL Web Ontology Language Overview. W3C Recommendation (2004), <http://www.w3.org/TR/owl-features/>
14. Osiński, S., Weiss, D.: Carrot2: Design of a flexible and efficient Web information retrieval framework. In: Szczepaniak, P.S., Kacprzyk, J., Niewiadomski, A. (eds.) *AWIC 2005*. LNCS (LNAI), vol. 3528, pp. 439–444. Springer, Heidelberg (2005)
15. Prud’hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF. W3C Recommendation (2008), <http://www.w3.org/TR/rdf-sparql-query/>
16. Sirin, E., Parsia, B., Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics* 5(2), 51–53 (2007), <http://dx.doi.org/10.1016/j.websem.2007.03.004>

Protégé Based Environment for DL Knowledge Base Structural Analysis

Mariusz Chmielewski and Piotr Stapor

Computer Science Department,
Cybernetics Faculty, Military University of Technology,
Kaliskiego 2, 00-908 Warsaw, Poland

Abstract. Structural analysis of knowledge bases improves process of obtaining additional hidden information and provides new means of information discovery. In this paper we propose a method of structural analysis and characterise in details developed toolkit as a Protégé extension. The aim of method itself is to identify hidden relationships using structural weighted graph analysis, applied for both terminology and instance base. In order to evaluate new relationships between instances, we have chosen a set of measures that consider the analysis of both vertices and links, thus providing new semantic information for identified associations. Developed environment concludes both approaches (logic based and graph based) in order to apply reasoning techniques, graph based algorithms and measures for inferring important analytical information. The scope of paper provides also application areas of such method in terms of data mining techniques for crisis identification and multi-criteria data analysis.

Keywords: semantic association, semantic similarity, ontology, OWL, description knowledge.

1 Introduction

Knowledge bases exploiting the Semantic Web [1] idea are becoming more and more competitive to legacy databases in terms of network data processing and analytical capabilities. The utilization of semantic networks as a mean of data representation, fused with the power of Description Logic [2] and reasoning mechanisms introduces enhancements in the field of automatic information discovery. Already available DL reasoners (ex. Pellet, Keon2, HermiT) and rule-based engines (ex. Jess) expose the abilities of instance classification process based on consistent rules and model validation. Protégé OWL [13], among many modelling features, offers a set of analytical functions such as ontology metrics and model visualisation, but at the same time it lacks algorithms for structural KB analysis. This gap can be filled by described Terrana plug-in, which concentrates mainly on applying ontology and instance base evaluation techniques for hidden information discovery. Term hidden information corresponds to all indirect relations between concepts or instances added after process of multi-criteria semantic association analysis. Evaluation of model characteristics is

performed on weighted multi-graph representation. Terrana implements model transformation rules to create multi-graph which reflects ontology or instance base elements connected with structural relations. Developed model is further used as a basis for evaluating characteristics of nodes and links considering both taxonomical and axiomatic information. Detailed information on transformation process and its accuracy and will be discussed later.

It is crucial to state that multi-criteria ranking process is achieved by introducing relevant ontology (concept, relationship) and instance base metrics, which provide evaluation of a wide spectrum of characteristics. Sources [9][12][11] distinguish variety of graph based factors which concentrate mainly on structure assessment, which we consider but also modify to drain semantics. Algorithms concentrate on element importance depending on centrality measures, such as: degree, connectivity, diameter etc. The ability to find hidden associations, combined with a set of algorithms for asserted relations linkage and evaluation is essential.

2 Semantic Model Formal Definition

In order to understand ranking process we should lay out the theoretical model on which method and software rely. We start with altered definition of ontology [6] formulated as:

$$O = \langle C, R_C, R_R^H, R_R^I, A^O \rangle \quad (\text{Eq.1})$$

where C is a set of identified unique concepts, R_C – is a set of relations between defined concepts and A^O - is a set of axioms defined for ontology O . Set $R_C = H_C \cup S_C$ is additionally divided based on the characteristics of relations to set S_C structural relations and hierarchical H_C relations used to organise concept taxonomy. R_R^H is a set of relations between elements of R_C identifying relation taxonomy and R_R^I is a set of relations between elements of R_C identifying specific semantics for chosen relations used to express inverse relations.

Using presented ontology O definition, there can be formed an instance base (data level knowledge base which is used to store the instances of elements defined on the conceptual level), defined as:

$$IN^O = \langle I_C^O, I_{R_C}^O, V_C^O, V_{R_C}^O \rangle \quad (\text{Eq.2})$$

, where: I_C^O contains instances of concepts C , $I_{R_C}^O$ contains instances of relations R_C in a given ontology O . Elements of instance base can be defined as follows:

$I_C^O = \bigcup_{c \in C} Inst_C^O(c)$, $Inst_C^O(c)$, identifies set of all instances for given concept $c \in C$;

$I_{R_C}^O = \bigcup_{r \in R_C} Inst_{R_C}^O(r)$, identifies set of all instances of relation $r \in R_C$ which meet

$Inst_{R_C}^O(r) = \{(x_i, x_j) \in I_C^O \times I_C^O : r = (V_C^O(x_i), V_C^O(x_j)) \wedge r \in R_C\}$;

$V_C^O : I_C^O \rightarrow 2^C$ is a classifier function which reflects possible types of instances I_C^O ;

$V_{R_C}^O : I_{R_C}^O \rightarrow 2^{R_C}$ is a relations classifier function, identifying a set of relation types for a chosen relation instance $I_{R_C}^O$;

The term semantic model shall, in further part of this paper, be understood as a pair:

$$M^{Sem} = (O, IN^O) \tag{Eq.3}$$

Ontology structure for further structural analysis will be defined as weighted multi-graph:

$$\overline{O} = \left\langle \Omega, \{f_k(c)\}_{k \in \{1, \dots, LF\}, c \in C}, \{g_l(r)\}_{l \in \{1, \dots, LG\}, r \in R_C} \right\rangle \tag{Eq.4}$$

where Ω is ontology graph structure on which we define families of functions for nodes-concepts $\{f_k(c)\}_{k \in \{1, \dots, LF\}, c \in C}$ and links-relations $\{g_l(r)\}_{l \in \{1, \dots, LG\}, r \in R_C}$.

Multigraph $\Omega = \langle C, R_C, P \rangle$, contains nodes C , edges R_C and triple relation $P = \{ \langle c_i, r, c_j \rangle : c_i, c_j \in C \wedge r \in R_C \}$, fulfilling conditions:

$$\forall_{r \in R_C} \exists_{\langle c_i, c_j \rangle \in C \times C} \langle c_i, r, c_j \rangle \in P \quad \forall_{r \in R_C} \forall_{\substack{c_i, c_j \in C \\ c_k, c_l \in C}} \left\{ \left[\langle c_i, r, c_j \rangle \in P \wedge \langle c_k, r, c_l \rangle \in P \right] \Rightarrow \left[(c_i = c_k) \wedge (c_j = c_l) \vee (c_i = c_l) \wedge (c_j = c_k) \right] \right\}$$

Weighted multi-graph elements stand for:

$f_k : C \rightarrow val_k^c$ is the k -th function described on the multi-graph's nodes (concepts), $k = 1, \dots, LF$, (LF – number of f functions); val_k^c is a k -th set of values describing concepts, $g_l : R_C \rightarrow val_l^{R_C}$ – the l -th function described on the multi-graph's links (relations), $l = 1, \dots, LG$ (LG – number of h functions), $val_l^{R_C}$ is a l -th set of values describing relations.

Using previous definitions we can formulate an instance network structure model which is used to store the instances of ontology elements, providing data level description:

$$\overline{IN^O} = \left\langle \Phi, \{\varphi_h(x)\}_{h \in \{1, \dots, LH\}, x \in I_C^O}, \{\phi_l(y)\}_{l \in \{1, \dots, LL\}, y \in I_{R_C}^O} \right\rangle \tag{Eq.5}$$

where Φ is an instance base structure multi-graph $\Phi = \langle I_C^O, I_{R_C}^O, \overline{P} \rangle$, which contains nodes I_C^O , edges $I_{R_C}^O$ (interpretation same as IN^O) and triple relation defined as

$\overline{P} = \{ \langle x_i, y, x_j \rangle : x_i, x_j \in I_C^O \wedge y \in I_{R_C}^O \}$, fulfilling conditions:

$$\forall_{y \in I_{R_C}^O} \exists_{\langle x_i, x_j \rangle \in I_C^O \times I_C^O} \langle x_i, y, x_j \rangle \in \overline{P} \quad \forall_{y \in I_{R_C}^O} \forall_{\substack{x_i, x_j \in I_C^O \\ x_k, x_l \in I_C^O}} \left\{ \left[\langle x_i, y, x_j \rangle \in \overline{P} \wedge \langle x_k, y, x_l \rangle \in \overline{P} \right] \Rightarrow \left[(x_i = x_k) \wedge (x_j = x_l) \vee (x_i = x_l) \wedge (x_j = x_k) \right] \right\}$$

on which we define families of functions for nodes-concept instances $\{\varphi_h(x)\}_{h \in \{1, \dots, LH\}, x \in I_C^O}$, edges-relation instances $\{\phi_l(y)\}_{l \in \{1, \dots, LL\}, y \in I_{R_C}^O}$. Respectively

$\varphi_h : I_C^O \rightarrow val_h^c$ is the h -th function described on the multi-graph's nodes (instances of

concepts), $h = 1, \dots, LH$ (LH – number of ϕ functions), val_h^{lc} - is a h -th set of values describing concept instances, $\phi_l : I_{R_c}^O \rightarrow val_l^{rc}$ is the l -th function described on the multi-graph's links (instances of relations), $l = 1, \dots, LM$ (LM –number of ϕ functions), val_l^{rc} is a l -th set of values describing relation instances.

Having multi-graph definitions we can describe semantic paths definitions for ontology structure \overline{O} and instance base structure \overline{IN}^O and provide interpretations for those elements (we consider both: trails (undirected) and path (directed)). Semantic connectivity (concept connectivity, instance connectivity respectively) are graph like chains (undirected) or paths (directed) permitted by both ontology and instance base definitions:

$$S_{con}^O(c_i, c_j) = (c_i, r_k, c_{i+1}, \dots, c_{j-1}, r_l, c_j), c_i, \dots, c_j \in C, r_k, \dots, r_l \in R_C$$

$$S_{con}^{IN}(x_i, x_j) = (x_i, y_k, x_{i+1}, \dots, x_{j-1}, y_l, x_j), x_i, \dots, x_j \in I_C^O, y_k, \dots, y_l \in I_{R_c}^O.$$

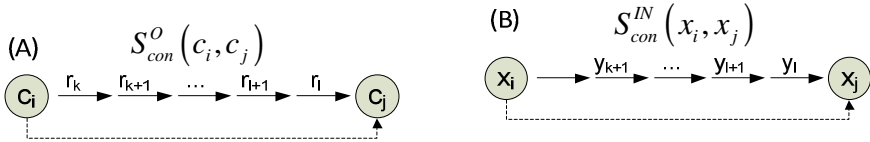


Fig. 1. Semantic connectivity (A) for concepts, (B) for instances

3 Terminology and Instance Base Measures

Some valuable information can be obtained by studying the ontology or instance base multi-graphs depending on proper vertices and links measure definitions. We base our assumptions on several observations:

- semantic model consist of taxonomical, structural and axiomatic information;
- each of those aspects can be evaluated using consistent approach which mainly require custom build structure (similar to presented earlier);
- ranking method must consider multi-criteria evaluation due to emphasised analysis approach;

As an example we can call in $Rank_{st}^O(c) = \frac{1}{k^{deg(c)}}$, $c \in C, k > 1$, a degree of a vertex

determines the informational usefulness of an ontology concept depending on the connection popularity of such node. In order to find and estimate the information value of the semantic associations, Terrana plug-in supplies user with following ranking factors: Semantic Link Relevance (SLR), Concept Clustering Coefficient (CCC), Concept Taxonomical Depth Measure (CTDM), and Semantic Concept Connectivity Measure (SCCM). Further part of this paper exploits in detail the aim of above mentioned measures and their usage in the field of hidden relation analysis.

Semantic Link Relevance has been introduced in the [9] and serves the purpose of determining the importance of existent or nonexistent link between two individuals or concepts. While small SLR value may indicate, that the link has little importance (and thus should be removed from semantic model), the large enough value may be a reason for “materializing”, the nonexistent (*potential* in [9]) link. The SLR parameter can also be used to make the semantic model analysis process more effective, by pointing the links in ontology or instance base multi-graphs that require more attention. SLR is expressed as the ratio of common neighbors to all neighbors owned by a pair of either individuals or concepts. Based on definitions of Link Relevance [9] we redefine measure in respect of ontology and instance base structures:

$$S_{rel}^O(c_i, c_j) = \frac{|N(c_i, c_j)|}{|T(c_i, c_j)|} \wedge c_i, c_j \in C, \quad S_{rel}^{IN}(x_i, x_j) = \frac{|N(x_i, x_j)|}{|T(x_i, x_j)|} \wedge x_i, x_j \in I_C$$

where, having regard to presented multi-graph model definitions on generic level:

$$S_{rel}(v_i, v_j) = \frac{|N(v_i, v_j)|}{|T(v_i, v_j)|} \quad \begin{aligned} N(v_i, v_j) &= \{v_k \mid v_k \text{ is linked to } v_i \wedge v_j, v_k \neq v_i, v_k \neq v_j\} \\ T(v_i, v_j) &= \{v_k \mid v_k \text{ is linked to } v_i \vee v_j, v_k \neq v_i, v_k \neq v_j\} \end{aligned}$$

Example 1. Calculating Semantic Link Relevance for (instance base)

where multigraph vertices $v \in I_C^0 \wedge e \in I_{R_C}^0$
 $b = \text{Critical infrastructure building } b, \quad t = \text{terrorist organization } t$
 $N(b, t) = \{Kabul, ARB \text{ org.}, \text{explosive materials}\}$
 $T(b, t) = \{\{bank\} \cup N(a, b)\}$

$$S_{rel}^{IN}(b, t) = \frac{|N(b, t)|}{|T(b, t)|} = \frac{3}{4} = 0.75$$

High Semantic Link Relevance measure indicates existence of asserted linkage between two given instances: critical financial infrastructure bank - building b and terrorist organization t . Such measure can be used to identify if additional relation should be introduced to link entities on terminological level (concepts).

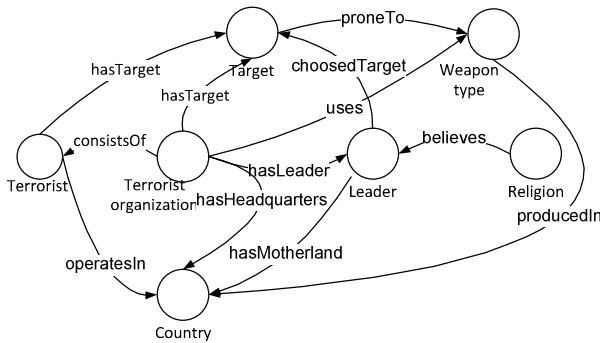
Considering the usefulness of SLR measure, there is also a need of defining a method of vertex importance computation. For this purpose, Terrana implements, Concept Clustering Coefficient (further referred as CCC) evaluation algorithm. In [9],

authors define CCC for an ontology with the help of formula: $CCC(c_i) = \frac{E(c_i)}{k_c(c_i)[k_c(c_i)-1]/2}$, where $c_i \in C$ in the defined model, $E(c_i)$ represents

links quantity between the nearest neighbors of vertex c_i and $k_c(c_i) = \text{deg}(c_i)$. In other words, the CCC value of a certain node c_i can be expressed as the ratio of number of existing connections between c_i neighbors divided by the quantity of all allowable connections between c_i neighbors.

The representation of ontology as a multi-graph implies the slight $CCC(c_i)$ adjustment which relays on large number of possible links existence. Algorithm treats any connections quantity between two neighbors with the same direction as one (as a result, the maximum number of links, taken into account during CCC evaluation, between any pair of vertexes is two). Additionally, because the model used in Terrana is directed multi-graph, the removal of division by two in denominator is implied. Alternatively it is possible to solve this issue, by constructing a multi-graph’s skeleton first, and then utilizing it in the process of CCC evaluation. Based on [9] we may assume that CCC greater than certain threshold, may be used to evaluate importance of node in terms of concept connectedness allowing us to infer that concept is worth considering in further analysis.

Example 2. Calculating Concept Clustering Coefficient example



$to = \text{Terrorist Organization}$, $wt = \text{WeaponType}$

$Neigh(c_i)$ – neighbors of given c_i concept

$Neigh(to) = \{ \text{Terrorist group, Target, Leader, Country, wt} \}$

$$k_{to} = |Neigh(to)| = 5$$

Existing connections between to neighbors $\{hasTarget, operatesIn, hasMotherland, choosesTarget, hasTarget, proneTo, producedIn\}$

$$\because k_{to}(k_{to} - 1) = 20 \wedge E_{to} = 7 \therefore CCC(to) = 0.35$$

$$\because k_{wt}(k_{wt} - 1) = 6 \wedge E_{wt} = 2 \therefore CCC(wt) = 0.33$$

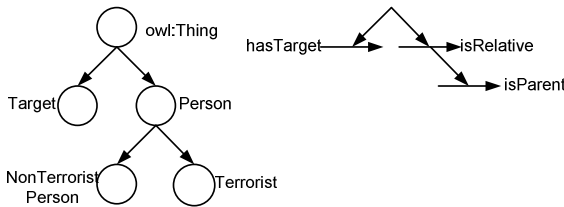
From the above, we may conclude, that for provided semantic model, the knowledge of used weapon type is less useful, than information about the terrorist organization which uses it.

3.1 Taxonomical Measures

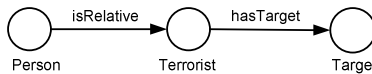
The taxonomical measures rely on ontology hierarchical relations providing the concepts' information load, stored in form of generalization-specialization tree. In Protégé, the taxonomy is stretched between owl:Thing (domain) and owl:Nothing (empty set) concepts. The deeper the placement within the taxonomy, the more detailed concept is (thus carrying more valuable semantic information).

Semantic association evaluation method utilizes the mentioned earlier hierarchy property. The informational level value for every concept (vertex) or relation (link), can be calculated using Concept Taxonomical Depth Measure (CTDM) formula: $I_{val}(c) = \frac{lvl(c)}{H(c)}$, where $lvl(c)$ function returns the level of c element in defined taxonomy, and $H(c)$ is the number of levels of whole hierarchy. For either classes or properties instances, we can determine their usefulness by evaluating $I_{val}(V_C^O(x))$ or $I_{val}(V_{RC}^O(y))$ respectively ($x \in I_C^O, y \in I_{RC}^O$). Those functions classify the instance thus helping to incorporate taxonomical measures for given instance. Having evaluated values for model elements, it is possible to estimate a given semantic association, expressed as simple acyclic graph path, as a following product $\prod_{a \in A_e} I_{val}(a)$, where A_e is a set of vertexes and links belonging to investigated association.

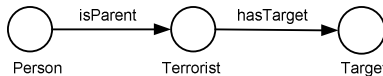
Example 3. Definition of concepts and relations taxonomies



In order to estimate informational value of two semantic associations, we will use the presented above formula:



$$\prod_{a \in A_e} I_{val}(a) = I_{val}(Person) \cdot I_{val}(isRelative) \cdot I_{val}(Terrorist) \cdot I_{val}(hasTarget) \cdot I_{val}(Target) = 0.5 \cdot 0.5 \cdot 1.0 \cdot 0.5 \cdot 0.5 = 0.0625$$



$$\prod_{a \in A_e} I_{val}(a) = 0.125$$

As shown, the more precise (specific) are the elements of studied association, the more useful they are. The information about terrorist's parent is more valuable than one about his (perhaps distant) relative.

3.2 Semantic Connectivity Based Measures

In order to improve Terrana's semantic model capabilities, we have chosen additional taxonomical measures [11], which rely on aggregated semantic connectivity S_{con}^O weighted measure:

- **Concept rank** $t_C^{rank}(c_i) = \frac{1}{h^{depth^C(c_i)}}$, where $h > 1$ and $depth^C(c_i) = lvl(c_i)$ returns the depth of a concept c_i in ontology concepts hierarchy tree.
- **Concept instance rank** $t_{IC}^{rank}(x_k) = t_C^{rank}(V_C^O(x_k))$.
- **Relationship rank** $t_R^{rank}(c_i, c_j) = \frac{1}{h^{depth^R(c_i, c_j)}}$, $h > 1$. $depth^R(c_i, c_j) = lvl(r(c_i, c_j) \in R_C)$ returns the depth of a property in an ontology properties hierarchy tree.
- **Direct relationship rank** $t_{RC}^{rank}(x_k, x_l) = t_R^{rank}(V_C^O(x_k), V_C^O(x_l))$

Considering above definitions, the final way of ranking given semantic association is as follows [11][9]:

$$t_{con}^{rank}(c_m, c_n) = t_{con(C)}^{rank}(c_m, c_n) \times t_{con(R)}^{rank}(c_m, c_n) \quad \text{where:}$$

$$t_{con(C)}^{rank}(c_m, c_n) = \prod_{c' \in S_{con}^O(c_m, c_n)} t_C^{rank}(c')$$

$$t_{con(R)}^{rank}(c_m, c_n) = \prod_{(c', c'') \in S_{con}^O(c_m, c_n)} t_R^{rank}(c', c'')$$

Example 4. Estimating semantic association

Let $s = S_{con}^O(\text{NotTerroristPerson}, \text{Target})$ and $h = 2$

$$t_{con(C)}^{rank}(\text{NotTerroristPerson}, \text{Target}) = 0.25 \cdot 0.5 \cdot 0.25 \cdot 0.5 = 0.01562$$

$$t_{con(R)}^{rank}(\text{NotTerroristPerson}, \text{Target}) = 0.25 \cdot 0.125 \cdot 0.5 \cdot 1 = 0.01562$$

$$t_{con}^{rank}(\text{NotTerroristPerson}, \text{Target}) = 0.01562 \cdot 0.01562 \approx 0,00024$$

4 Multi-criteria Association Evaluation

Depending on the analyst requirements, presented method is able to provide adjustments for evaluating structural measures and ranking semantic associations.

Each measure is connected with a certain evaluation criteria. For example: $m_{hier}^c = \frac{lvl(c)}{h^{depth^C(c)}}$, where $c \in C$ uses specialization level as its criterion. $m_{dist}^{c_i, c_j}$ would promote classes less edges apart from certain important concept c_j , and m_{deg}^c would judge the importance basing on number of neighbours of a given c vertex (degree). The problem of choosing the right value arises - how to interpret the set of results from different methods. To solve this problem, we recommend the use of multi-criteria approach:

- define measures m_i $i \in \overline{1, n} = \{1, 2, 3, \dots, n\}$, where $n \in \mathbb{N}$, additionally for each measure $m_i \in \langle 0, 1 \rangle$
- define weights w_i $i \in \overline{1, n}$, such that $\sum_{i=1}^n w_i = 1$;
- define vector M , such that $M^T = [m_1 \ m_2 \ \dots \ m_n]$;
- define vector W , such that $W^T = [w_1 \ w_2 \ \dots \ w_n]$;
- count the aggregated measure $M_{agg} = W \cdot M = W^T M = \sum_{i=1}^n w_i m_i$

Using prepared evaluation tools presented in this work so far, it is possible already to appraise usefulness of: concepts, individuals, properties, instances of properties, and in the end semantic associations. Additionally, environment provides information concerning the whole structure of the multi-graph. The summary for testbed instance base has been placed beneath.

Table 1. Instance data evaluation based on proposed graph based measures ranking importance on instance base nodes (facts) and their semantic importance inside KB

Order:65 Size: 55	Degree in,out	CCC	Betweenness	Closeness	Eigenvector Centrality	PageRank (0.15)	HITS hub (0.5)	HITS (0.5)
<i>September11Attacks</i>	0 / 8,8	0.0	0	1.7619	0.0123	0.0129	0.1184	0.0060
<i>HajjKhalilBanna</i>	1 / 4,5	1.0	0	0.8000	0.0153	0.0151	0.4348	0.1763
<i>PlaneHijack</i>	1 / 0,1	0.0	0	0.0000	0.1377	0.0143	0.0161	0.0364
<i>SuicideAttacks</i>	1 / 0,1	0.0	0	0.0000	0.0137	0.0143	0.1607	0.0364
<i>MilitaryBuildings</i>	0 / 0,0	0.0	0	0.0000	0.0122	0.0129	0.1607	0.0060
<i>AbuNidalOrg.</i>	2 / 3,5	0.3	10	1.5454	0.0459	0.0368	0.0781	0.1928
<i>AlQaeda</i>	2 / 7,9	0.0	18	1.3077	0.0162	0.0164	0.0540	0.1555
<i>OsamaBinLaden</i>	1 / 6,7	0.0	10	1.3846	0.0146	0.0149	0.4642	0.0199
<i>Hezbollah</i>	1 / 6,7	0.0	11	0.8571	0.0151	0.0154	0.0510	0.0904
<i>PalestineLiberationOrg.</i>	2 / 1,3	1.0	0	2.3636	0.0306	0.0255	0.0642	0.1964

Link Name	hasLeader	hasGender	hasConnectionWithTerroristOrg	hasEthnicity
CLR	0.1429	0.1667	0.5000	0.1429

5 Environment Architecture

We have chosen to implement all mechanisms of association discovery in Protégé due to its modelling and ontology processing capabilities. This approach has been motivated to maximize reuse of already developed software components such as: knowledge base operations, inference mechanisms integration, modelling language processing etc. Plugin capabilities delivers different semantic model usage compared to those directly found in Protégé OWL, which concentrates mainly on logical and consistent model design.

Review of those capacities confronted with association analysis method needs, helped us to identify requirements for future extensions addressing:

- weighted graph transformation mechanisms;
- domain terminology evaluation (classes, properties);
- instance base data assessment (individuals) and propagation of found premises to ontology model;
- complex path evaluation for terminology and instance base;

Reuse of available components concentrated on using Jess SWRL rule processing engine and Pellet as DL reasoner. The solution relies heavily on graph processing framework provided by JUNG library – used for ontology and instance base model representations, their visualization and invariants evaluation.

Model transformer component scans the structure of supplied by Protégé knowledge base and maps it to a JUNG based multi-graph according to prepared rules. Further processing includes graph invariants computation using JUNG library supplied algorithms (ex. Betweenness, Closeness, Eigenvector Centrality, PageRank, HITS) and our internal implementation of (Concept Clustering Coefficient, Concept Link Relevance).

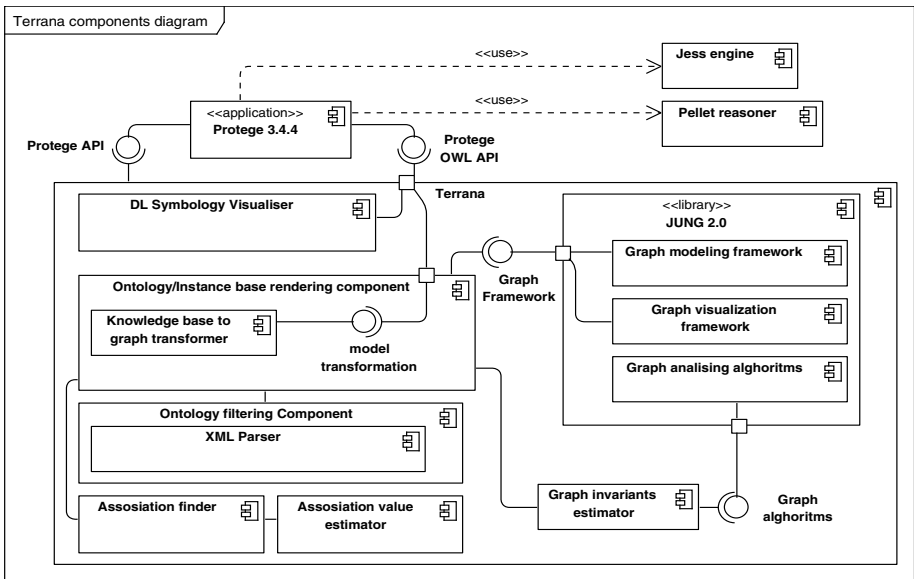


Fig. 2. Developed environment components and used external libraries

Multi-graph representations are utilized by semantic connectivity algorithm which concentrates on searching for a simple acyclic path (or all such paths) between given two concepts or individuals. The association value estimator computes the importance/informational value of such discovered connection using a selected evaluation method.

6 Summary

Information gathering in knowledge base may introduce new approach in analysis while applying flexible graph structures. Presented approach lay foundations and exploits many possible applications for knowledge base validation and data mining techniques. Development of new, more effective ontology measures is followed by a research explaining how those measures can assess instant data, identify hidden relationships and evaluate terminology design. Implemented method concentrates on distinct quantitative analysis of ontologies and instance bases for information discovery and validation. Method utilises multi-criteria expert tuned approach in order to extend structural KB elements evaluation based on known node importance measures applied on the ground of semantic models.

Designed environment has been applied for solving terrorism related problems using adaptive information gathering and to semantic model representation mapping techniques. Processing of such knowledge base may be performed by logical reasoning and structural analysis, which mainly concentrates on: (a) finding hidden transitive relations between chosen instances; (b) classification of concept instances based on gathered characteristics and defined rules, (c) prediction of possible events (based on structure of certain vertexes and relations between them); (d) modeling complex relationships between main instances.

References

1. Davies, J., Studer, R., Warren, P.: *Semantic Web Technologies: Trends and Research in Ontology-based Systems*. John Wiley & Sons, Chichester (2006)
2. Baader, F., McGuinness, D., Nardi, D., Patel-Schneider, P.: *The Description Logic Handbook: Theory, implementation, and applications*. Cambridge University Press, Cambridge (2007) ISBN 0521876257
3. Cardoso, J.: *Semantic Web Services: Theory, Tools and Applications*, . IGI Global (March 2007) ISBN-10: 159904045X
4. Segaran, T.: *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. O'Reilly Media, Sebastopol (2007) ISBN 0596529325
5. Staab, S., Studer, R.: *Handbook on Ontologies*. Springer, Heidelberg (2004) ISBN 3540408347
6. Gruber, T.R.: *Toward principles for the design of ontologies used for knowledge sharing*. In: *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic, Boston (1993) (in preparation)
7. Gruber, T.R., Olsen, G.R.: *An Ontology for Engineering Mathematics*
8. Chmielewski, M., Gałka, A., Jarema, P., Krasowski, K., Kosiński, A.: *Semantic Knowledge Representation in Terrorist Threat Analysis for Crisis Management Systems*. LNCS. Springer, Heidelberg (2009)
9. Barthelemy, M., Chow, E., Eliassi-Rad, T.: *Knowledge Representation Issues in Semantic Graphs for Relationship Detection*, UCRL-CONF-209845

10. Chmielewski, M., Kasprzyk, R.: Multi-criteria semantic association ranking based on instance knowledgebase analysis for criminal organisation identification. In: Euro 2010 (2010)
11. Ge, J., Qiu, Y.: Concept Similarity Matching Based on Semantic Distance. Faculty of Computer and Information Science. Southwest University Chongqing, China
12. Baur, M., Benkert, M.: Network comparison. In: Brandes, U., Erlebach, T. (eds.) Network Analysis. LNCS, vol. 3418, pp. 318–340. Springer, Heidelberg (2005)
13. Protégé OWL webpage,
<http://protege.stanford.edu/overview/protege-owl.html>

Fuzzy Reliability Analysis of Simulated Web Systems

Tomasz Walkowiak and Katarzyna Michalska

Institute of Computer Engineering, Control and Robotics,
Wroclaw University of Technology,
ul. Janiszewskiego 11/17, 50-372 Wroclaw, Poland
{tomasz.walkowiak, katarzyna.michalska}@pwr.wroc.pl
<http://www.pwr.wroc.pl>

Abstract. The paper presents a new approach to reliability analysis of Web systems using fuzzy logic. Since the reliability parameters of the Web system are often approximated by experts we propose to analyze Web systems using fuzzy logic methodology. We assess the reliability of the system by accumulated down time. Fuzzy logic based reliability analysis, as well as Web system modeling and simulation are presented. Moreover, results of numerical experiment performed on a test case scenario using proposed methodology are given.

Keywords: fuzzy analysis, web systems, reliability, fault model.

1 Introduction

Decisions related to complex Web systems ought to be taken based on different and sometimes contradictory conditions. The reliability - in a wide sense - is one of the main factors, which are taken into consideration during a decision process. The reliability maybe isn't the most important factor but is of a great weight as a support criterion. So quantitative information related to the reliability characteristics is important and can be used as a decision-aided system if it is necessary to discuss different economic or quality aspects [4], [17], [6], [16], [7]. The typical models used for reliability analysis are mainly based on Markov or Semi-Markov processes [1] which are idealized and it is hard to reconcile them with practice.

The typical reliability analysis uses very strict assumptions related to the life or repair time and random variables distributions of the analyzed system elements. That's way we propose to use Monte-Carlo [5] approach for reliability analysis. Moreover, modern computer devices are characterized by very screwed up reliability parameters - sometimes they are completely unmeasured or the population of analyzed elements is too limited to find such general information. So - the reliability parameters are fixed based on the experts' experience. This fact is the reason why fuzzy approach to reliability parameters description proposed by authors for discrete transport systems [8], [9] could also be applied for Web systems.

The approach is based on the assumption that one could calculate some metrics for a given system configuration, including reliability parameters of the system. Next, the reliability parameters are described using linguistic variables. Such system could be understood as multiple input and one output fuzzy system. And by a usage of fuzzy operators one could calculate the overall metric value as a fuzzy variable or after a defuzzification as a crisp value. The crucial point for such approach is the calculation of system metric. In the paper we propose to calculate the mean accumulated down time (section 3) by a usage Monte-Carlo [5] based web system simulator [15] [14].

2 Web Systems

2.1 System Model

The analyzed class of Web systems is described on three levels. On the top level, it is represented by interacting service components. At the bottom, technical level it is described by hosts, on which the services are located. The intermediate level describes the mapping between the other two. Service components (interacting applications) are responsible for providing responses to queries originating either from the system users or from other service components. While computing the responses, service components acquire data from other components by sending queries to them. The system comprises of a number of such components. The set of all services comprises a Web system. Communication between Web services works on top of Internet messaging protocols. The communication encompasses data exchange using the client-server paradigm. The over-all description of the interaction between the service components is determined by its choreography, i.e. the scenarios of interactions that produce all the possible usages of the system. The service components interact with each other in accordance with the choreography. As the result, there are logical connections between service components. The service component is realized by some technical service, like for example Apache, Tomcat or MySQL server. The technical service is placed on some hosts. The assignments of each service components to a technical service and therefore to a host gives the system configuration. We assumed [15] that aspects of TCP/IP communication throughput aspects for modern Web based systems (not including video streaming systems) could be omitted.

2.2 Functional Model

The performance of any Web system has a big influence on the business service quality. It has been shown [11] that if user will not receive answer from the system in less than 10 seconds he/she will probably resign from active interaction with the system and will be distracted by other ones. Therefore, the most important part of the system model is an algorithm that allows to calculate how long a user request will be processed by a system. The processing of a user request is done by service components according to a given choreography, so the overall processing time could be calculated as equal to time needed for communication

between hosts used by each service component and the time of processing tasks required by each of service components. Since, we omitted TCP/IP aspects, the communication time was model be a random value (with truncated normal distribution). In case of task processing time the problem is more sophisticated. It is due to a fact that the processing time depends on the type of a task (its computational complexity), type of a host (its computational performance) on which a task is executed and a number of other tasks being executed in parallel. And this number is changing in a time during the system lifetime. Therefore, it is hard to use any of analytic methods to calculate the processing time. That is way we used the simulation approach that allows to monitor the number of executed tasks on each host during the simulation process. Having, calculated the time of processing a user request one could use it for assigning a request to be not correctly handled if it exceeds a time limit (10 seconds was used in presented experiments). There, could be also other sources of not correctly requests. The communication protocols (like HTTP) as well as Web services (for example JSP) have built-in timeouts. If a request is not finished within a given time limit (in most cases it could be set by one of configuration parameters) it is assumed to be failed. The other reason of not correctly handled requests in Web systems is a limit to a number of tasks handled by a technical service (i.e. Tomcat) at the same time. Since most of the user tasks consist of a sequence of requests, if one from the sequence fails the whole user request is assumed to be not correctly handled. All these phenomenons have been included in our Web system model [15]. The other sources of not correctly answered requests are hardware and software failures.

2.3 Fault Model

The previous section introduced failures as a result of system functionality, i.e. a result of time-outs and maximum number of requests. We propose to extend failures to represents Web system faults which occur in a random way. Of course, there are numerous sources of faults in complex Web systems. These encompass hardware malfunctions (transient and persistent), software bugs, human mistakes, viruses, exploitation of software vulnerabilities, malware proliferation, drainage type attacks on system and its infrastructure (such as ping flooding, DDOS) [2], [3]. We propose to model all of them from the point of view of resulting failure. We assume that system failures could be modeled a set of failures. Each failure is assigned to one of hosts and represents a separate working-failure stochastic process. The proposed fault model takes into account different types of faults, like: viruses or host and operating system failures. The occurrence of failure is described by a random process. The time to failure is modeled by the exponential distribution. Whereas the repair time by truncated normal distribution. In simulation experiments described in the next section we consider two types of failures for each host: with full dysfunction of host and 98% downgrade of host performance. The first one, represents the results of a host or operation system failure. The second type of faults (with 0.98 downgrade parameter) model a virus or malware occurrence.

2.4 Programming Simulation

The above model was analyzed by means of computer simulation. A software package for Monte-Carlo simulation [5] has been developed by authors [15]. It is based on the publicly available SSF [10] simulation engine that provides all the required simulation primitives and frameworks, as well as a convenient modeling language DML [10] for inputting all the system model parameters. The simulation algorithm is based on tracking all states of the system elements. The state is a base for a definition of an event, which is understood as a triple: time of being happened, object identifier and state. Based on each event and states of all system elements rules for making a new event has been encoded in the simulation program. The random number generator was used to deal with random events, i.e. failures. By repeating the simulator runs multiple times using the same model parameters and obtains several independent realizations of the same process (the results differ, since the system model is not deterministic). These are used to build the probabilistic distribution of the results, especially the average measures.

3 System Metric

3.1 System availability

The quality of system in a given operational state could be described by the system availability. It is usually defined as the probability that the system is operational (provides correct responses) at a specific time. Assuming a uniform rate of requests in a specific time horizon (from $t - \Delta$ to t), the availability could be estimated as the ratio of number of requests correctly handled ($N_{OK}(t - \Delta, t)$) by the system over a number of all requests ($N(t - \Delta, t)$):

$$A(t) = \frac{N_{OK}(t - \Delta, t)}{N(t - \Delta, t)}. \quad (1)$$

3.2 Mean Accumulated Down Time

A good metric used for describing the system quality from the reliability point of view is the mean accumulated down time (MADT). It shows the number of hours a system is unavailable during a year. To define it, we need to state what does it mean that a web system is unavailable. We propose to use availability metric described above, and assume that a system for which availability at a given time is smaller than some threshold is in down state, in other case it is in operational state. Therefore, we could calculate the accumulated down time over one year, as:

$$ADT = \int_{t=0}^{1year} \mathbf{1}(A(t) < \theta) dt, \quad (2)$$

where: $\mathbf{1}()$ - is a binary function giving 1 for true argument, and 0 for false; θ - is a threshold that defines the level of require availability to assume that a system is in operational state.

The metric is illustrated in Fig. 1.

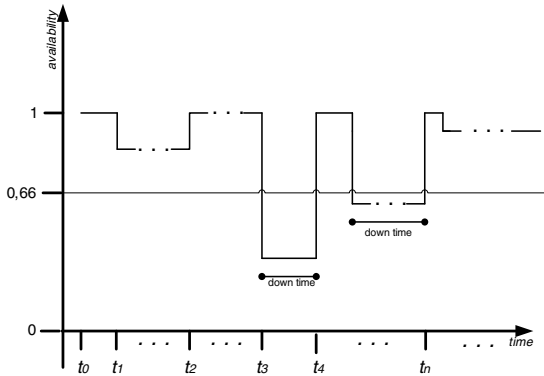


Fig. 1. Web system infrastructure - case study example

The above value is a random one, since it depends on number and types of failures that occurs during analyzed year, therefore in practice we use its expected value: a mean accumulated down time:

$$MADT = E \left[\int_{t=0}^{1year} \mathbf{1}(A(t) < \theta) dt \right]. \tag{3}$$

4 Fuzzy Analysis

4.1 Case Study

To show the possibilities of the proposed model we have analyzed a Web system presented in Fig. 2. It consists of three networks: one is a client network, other service provider networks (secured by a Firewall). System (testbed) is realizing simplified education system, that allows to register to the university database and assign to the available courses.

Few servers are used for a proper service realization: *StudentsDatabase* - information storage about students, *WebServer* - Web Page server, *Database* - Database server (data for this system can be stored on more than one database), *BackupDatabase*, *TimetableDatabase* - server responsible for keeping informations about available courses, *RegisterServer*. Education service and it's choreography is described in Fig. 3. First of all, students can look at the Web-Page

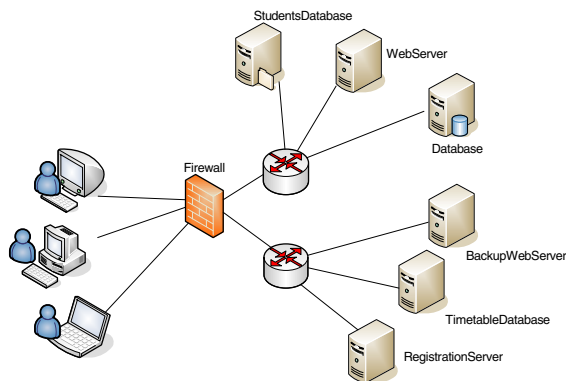


Fig. 2. Web system infrastructure - case study example

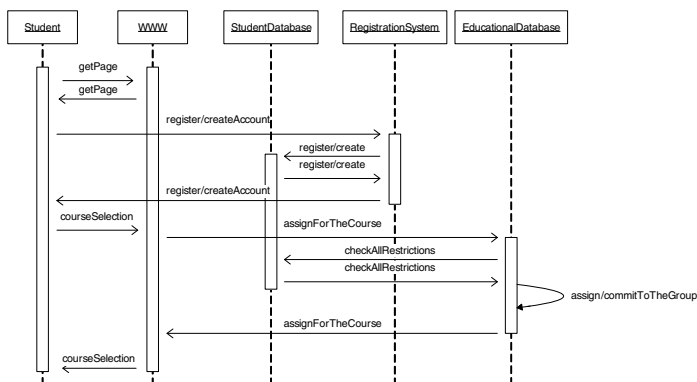


Fig. 3. Web system choreography - case study example

and log in or register. When a student is in a database many options are available, i.e. searching for the course or assignment to a specified course. If a student is register and is allowed to assign to the group, the scenario shown on Fig. 3 is realized as requested.

4.2 Fuzzy Reliability Parameters

We want to analyze the accumulated down time in a function of fuzzy representation of host reliability parameter: mean time of failures (fh) and mean repair time (rh). We are not analyzing the classical reliability values: intensities of failures but its inverse since we think that it is much easier for expert to express the failure parameter in time units [8]. Moreover, as it was described in section

2.3. we analyze the occurrence of virus and malware intrusions. It is described by mean time of virus occurrence (fv) and mean repair time (rv).

We propose to use a trapezoidal membership function for fuzzy representation of mean time of failures and repair time for host and virus failures. Let note it is as: $\mu_{type}()$, where $type$ is equal to f_h, r_h, f_v, r_v for mean time to host failure, host repair time, mean time to virus and virus repair time respectively. Assumption of the fuzzy membership function shape does not bound the analysis. One could use any other membership function and apply presented here methodology. For mean time of host failures, the four trapezoidal parameters of fuzzy membership function was set to (290,330,670,710) days. Today’s computer devices to not fail very often, that is way we consider a host failures mean time between one to two years. Faults that are related to viruses are more probable than a host failure, especially for systems that are exposed to attacks. Web systems are definitely in this group. Therefore, in our study the mean time to virus occurrence fuzzy trapezoidal parameters were set to (100,140,340,360). In case of repair time we use (4,8,32,48) hours for host repair time and (2,4,16,24) for virus repair time.

4.3 Fuzzy Mean Accumulated Down Time

In general we propose the analysis of mean accumulate down time in a function of realizability parameter:

$$MADT(fh, rh, fv, rv). \tag{4}$$

The values of this function are calculated using simulator tool mentioned in section **2.4.** with 10 users per second average input load. The above function creates a multiple input single output (MISO) fuzzy system with four inputs. Applying fuzzy operator like max and multiply one could calculate the output membership function, as follows:

$$\mu_{MADT}(t) = \underset{t=MADT(mh, rh, mv, rv)}{MAX} \{ \mu_{f_h}(fh) \cdot \mu_{r_h}(rh) \cdot \mu_{f_v}(fv) \cdot \mu_{r_v}(rv) \} \tag{5}$$

Results for the case study web systems are presented in Fig. **4a.**

4.4 Fuzzy Simplified Approach

The calculation of the above formula is complicated and time consuming. Therefore, we propose to calculate the output membership function based on L-R representation **12** of fuzzy variables and by approximation of resulting system fuzzy membership. We calculate the resulting output function value **(4)** only for characteristic points of trapezoidal fuzzy membership of input values. For our four dimensional space, it gives $2^4 = 16$ points for fuzzy membership function equal to 1 and similarly for membership function equal to 0. Among each of these two groups, minimum and maximum values of the output function values **(4)** is

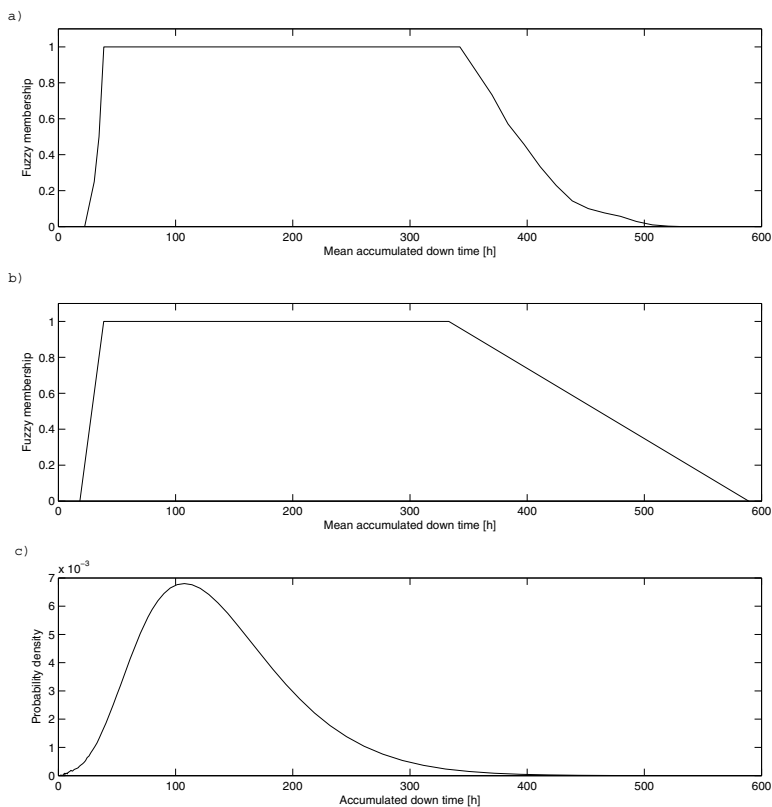


Fig. 4. Results: the accumulated down time presentation by a) fuzzy method b) fuzzy simplified method and c) probability density method

selected giving the resulting trapezoidal membership output function. Such representation guarantees a really simple and fast calculation of fuzzy output values. But this is only the rough approximation of fuzzy function of outputs values. The results for case testbed are shown in Fig. 4b.

4.5 Probability Density Method

We propose also the other way of final results presentation, based on probability density function estimation. Assuming that the fuzzy representation of mean time to failure and repair time (section 4.2) is a way of stating the probability of time to failure, we could calculate the overall gain probability density function using slightly modified kernel method (with Gaussian kernels). The modification is done by multiplication each kernel by the weighted fuzzy trapezoidal function. Based on I results of accumulated down time $ADT_i(fh, rh, fv, rv)$ achieved for different reliability parameter values (fh, rh, fv, rv) by computer simulation, the density function $f(t)$ could be approximated by:

$$f(t) = \frac{1}{h\sqrt{2\pi} \sum_{j,k,l,m=1}^{J,K,L,M} \mu_{f_{-h}}(fh_j) \cdot \mu_{r_{-h}}(rh_k) \cdot \mu_{f_{-v}}(fv_l) \cdot \mu_{r_v}(rh_m)} \cdot \sum_{i,j,k,l,m=1}^{I,J,K,L,M} \exp\left(-\frac{1}{2} \left(\frac{ADT_i(fh_j, rh_k, fv_l, rh_m) - t}{h}\right)^2\right) \cdot \mu_{f_{-h}}(fh_j) \cdot \mu_{r_{-h}}(rh_k) \cdot \mu_{f_{-v}}(fv_l) \cdot \mu_{r_v}(rh_m)$$

where h is a bandwidth parameter. It is set to optimal value based on maximal smoothing principle: AMISE - the asymptotic mean square error [13].

Results for the case study web systems are presented in Fig. 4c.

5 Conclusion

Summarizing, we proposed a method of Web system fuzzy analysis. It is based on fuzzy reliability representation and computer simulation which allow to soften the typical assumptions related to the system structure and to reliability parameters of web system elements. Using proposed solution Web system could be verified against quality requirements, what makes this approach a powerful tool for increasing system dependability and by that increasing satisfaction of the service user. Considering complexity of the Web system, we keep in mind that more and more parameters should be specified in a similarly manner. Taking into consideration the computation complexity of accumulated down time estimation (simulating the system for different operational states with different system parameters) we suggest to use approximation of system fuzzy membership function based on L-R representation (section 4.4). Researches in this area are still in progress, with respect to more complicated testbeds and larger data set.

In the future, we plan to extend our solution to take into account the diversity of accumulated down time which could be seen in results presented in Fig. 4c. The origin of the diversity is the randomness presented in Web systems (in our case mainly failures and repair process). And the presented fuzzy approach was based on the mean value of accumulated down time for each set of system parameters. Moreover, it would be interested to model the changing number of the system users by fuzzy values.

We hope, that presented approach could be a foundation for a new methodology of the reliability and quality analysis of Web Systems, which is much closer to the practice experience.

Acknowledgment. The presented work was funded by the Polish National Science Centre under contract no. 4759/B/TO2/2011/40.

References

1. Barlow, R., Proschan, F.: Mathematical Theory of Reliability. Society for Industrial and Applied Mathematics, Philadelphia (1996)

2. Chan, K.S., Bishop, J., Steyn, J., Baresi, L., Guinea, S.: A fault taxonomy for web service composition. In: Di Nitto, E., Ripeanu, M. (eds.) ICSOC 2007. LNCS, vol. 4907, pp. 363–375. Springer, Heidelberg (2009)
3. Conallen, J.: Building Web Applications with UML. Addison Wesley Longman Publishing Co., Amsterdam (2000)
4. Davila-Nicanor, L., Mejia-Alvarez, P.: Reliability improvement of Web-based software applications . In: Proceedings of Quality Software, QSIC 2004, September 8-9, pp. 180–188 (2004)
5. Fishman, G.: Monte Carlo: Concepts, Algorithms, and Applications. Springer, New York (1996)
6. Ma, L., Tian, J.: Web error classification and analysis for reliability improvement. Journal of Systems and Software 80(6), 795–804 (2007)
7. Martinello, M., Kaaniche, M., Kanoun, K.: Web service availability–impact of error recovery and traffic model. Safety, Reliability Engineering & System Safety 89(1), 6–16 (2005)
8. Mazurkiewicz, J., Walkowiak, T.: Fuzzy Reliability Analysis. In: ICNNSC 2002 – 6th International Conference Neural Networks and Soft Computing. Advances in Soft Computing, pp. 298–303. Physica-Verlag, Heidelberg (2003)
9. Mazurkiewicz, J., Walkowiak, T.: Fuzzy economic analysis of simulated discrete transport system. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 1161–1167. Springer, Heidelberg (2004)
10. Nicol, D., Liu, J., Liljenstam, M., Guanhua, Y.: Simulation of large scale networks using SSF. In: Proceedings of the 2003 Winter Simulation Conference, vol. 1, pp. 650–657 (2003)
11. Nielsen, J.: Usability Engineering. Morgan Kaufmann, San Francisco (1994)
12. Piegat, A.: Fuzzy Modeling and Control. EXIT Academic Publishing House, Warsaw (1999) (in Polish)
13. Silverman, B.W.: Density Estimation. Chapman and Hall, London (1986)
14. Walkowiak, T., Michalska, K.: Performance analysis of service-based information system with load balancer - simulation approach. Dependability of Networks, Oficyna Wydawnicza Politechniki Wroclawskiej, 155–168 (2010)
15. Walkowiak, T.: Information systems performance analysis using task-level simulator, pp. 218–225. IEEE Computer Society Press, Los Alamitos (2009)
16. Tian, J., Ma, L.: Web Testing for Reliability Improvement. In: Zelikowitz, M.V. (ed.) Advances in Computers, vol. 67, pp. 177–224. Elsevier, Amsterdam (2006)
17. Zhang, J., Zhang, L.-J.: Criteria analysis and validation of the reliability of Web services-oriented systems. In: IEEE International Conference on Web Services (ICWS 2005), pp. 621–628 (2005)

Using Multi-attribute Structures and Significance Term Evaluation for User Profile Adaptation

Agnieszka Indyka-Piasecka

Institute of Informatics, Wrocław University of Technology, Poland
agnieszka.indyka-piasecka@pwr.wroc.pl

Abstract. This contribution presents a new approach to the representation of user's interests and preferences. The adaptive user profile includes both interests given explicitly by the user, as a query, and also preferences expressed by the valuation of relevance of retrieved documents, so to express field independent translation between terminology used by user and terminology accepted in some field of knowledge. Procedures for building, modifying and using the profile, heuristic-based significant terms selection from relevant documents are presented. Experiments concerning the profile, as a personalization mechanism of Web search system, are presented and discussed.

1 Introduction

In today's World Wide Web reality, the common facts are: increasingly growing number of documents, high frequency of their modifications and, as consequence, the difficulty for users to find important and valuable information. These problems caused that much attention is paid to helping user in finding important information on Internet Information Retrieval (IR) systems. Individual characteristic and user's needs are taken under consideration, what leads to system personalization. System personalization is usually achieved by introducing user model into information system. User model might include information about user's preferences and interests, attitudes and goals [3], knowledge and beliefs [6], personal characteristics [7], or history of user's interaction with a system [12]. User model is called user profile in the domain of IR. The profile represents user's information needs, such as interests and preferences and can be used for ranking documents received from the IR system. Such ranking is usually created due to degree of similarity between user query and retrieved documents [8], [14]. In Information Filtering (IF) systems, user profile became the query during process of information filtering. Such profile represents user information needs, relatively stable over a period of time [1]. User profile also was used for query expansion, based on explicit and implicit information obtained from the user [5].

The main issue, in the domain of user profile for IR, is a representation of user information needs and interests. Usually user interests are represented as a set of keywords or a n -dimensional vector of keywords, where every keyword's weight at the vector represents importance of keyword to representing user interests [8], [9]. The approaches with more sophisticated structures representing knowledge about user's preferences are also described: stereotypes – the set of characteristics of a prototype

user of users groups, sharing the same interests [4], or semantic net, which discriminates subject of user interests by underlining the main topic of interests [2].

The approaches to determine user profile can be also divided into a few groups. The first group includes methods where user's interests are stated explicitly by the user in specially prepared forms or during answering standard questions [4], [8], or by an example piece of text, written by the user [11]. The second group can be these approaches, where user profile is based on the analysis of terms frequency in user queries directed to IR system [8]. There is an assumption for these methods that the interest of the user, represented by a term, is higher as the term is more frequent in the user query. Analysis of the queries with the use of genetic algorithms [13], reinforcement learning [18] or semantic nets [2] are extensions to this approach. The third group of approaches includes methods, where the user evaluates retrieved documents. From documents assessed as interesting by the user, additional index terms, describing user interests, are added to the user profile [4], [8].

Most of research, concerning user modeling for IR, acquire user information needs expressed directly by the user of the IR system. The severe difficulties in expressing the real information needs by the user are frequently neglected. The fact that user usually does not know, which words he should use to formulate his interests to receive valuable documents from the IR system is ignored. We claim that user can express own preferences by retrieved documents relevance valuation.

2 Analysis of Web System Answer

Documents of the answer include documents returned by Web search system. Not all documents found by the system pertain to the real user interest. We assume that when we ask a user to select among documents of the answer those which are more interesting for him, we can come closer in this way to the picture of the domain of his interest. In order to make the selection process realistic we assume that user is making only a binary choice: interesting vs. non-interesting, i.e. to point out some documents without further assessment, e.g., of a kind of strength of how they pertain to his needs.

For the needs of IR on the Web, a domain of user interest is represented and identified by keywords that originate and were extracted from the selected and relevant documents of the answer. An important element of the newly proposed representation of user interest, i.e. the user profile, is an analysis of answer documents. The process is aimed to identify these terms which are representative for relevant documents of the answer and on the other hand are key terms from domain of user interest. The ultimate goal of the answer documents analysis is identification of vocabulary used in a certain knowledge domain which is the domain of user interest and the construction of the user interest representation on the basis of the relevant documents.

2.1 Relevant Document's Terms Weighting

Each term from a document indexed in Web search system is assigned the weight d_i according to the *tf-idf* schema [17]. The weights allow for *index terms* identification

which describe properly the document content. A term t_i can occur in more than one relevant document. Thus the weight wz_i' is influenced also by the weights d_i of this term in each of the selected relevant documents of an answer.

2.2 Significant Terms Selection from Relevant Documents

Key terms, which are important for the domain of user interest, are automatically extracted from the relevant documents selected by the user. These terms are stored in the appropriate subprofile and next used during modification of some of the following questions asked by the user to the system.

Key terms chosen from the relevant documents selected are henceforth called *significant terms*. The proposed term selection method was inspired by the idea of discriminative terms [16], [19] and the cue validity factor [10]. Selected significant terms are introduced into a subprofile and than used to modify user query.

The proposed method of significant terms selection is performed on several steps. Assigning wz_i' weights to each term from relevant documents leads to set of significant terms tz_i from the relevant documents. A joint application of two term selection criteria is important novelty of our approach. In this method, weights of terms from the relevant documents together with the cue validity factor are combined to a two-step filter. A criterion obtained is a *weighted sum*. As an effect of combining two discussed method of weighting terms from relevant documents, among all terms belonging to the relevant documents, only the terms pertaining to vocabulary used in the domain of user interest are selected. A weight of term-candidate for inclusion into the set of significant terms is calculated as following:

$$wz_i = \alpha wz_i' + \beta cv_i \quad (1)$$

where values of factors α and β were chosen experimentally.

Selection of discriminative terms on the basis of a constant threshold is the technique often reported in literature and applied in IR, e.g. it was used for the authoritative collection of documents. According to this technique only terms with weight above some constant, pre-defined threshold are added to the selected group.

However collections of Web documents express substantially different properties. These collections are characterized by huge divergence of topics and large dynamics in time in relation to both: the number of terms and documents. In such collections, term significance, represented by term weight, is changing according to the modifications introduced into the collection, i.e. after adding new documents or altering already present ones. A typical method of discriminative terms identification, i.e. on the basis of the threshold expressed by the constant values, when applied to the Web collection would not produce an expected set of significant terms. So we propose multi-attribute criterion for significant terms identification. Values of such thresholds are constant, but defined on the basis of functions considering the dynamic of term weights in the Web collection.

The process of significant terms selection is performed in the following steps:

1. The user verifies the answer from Web search system by selecting relevant documents (binary selection).

2. The weight d_i is calculated for each term occurring in the relevant document. The weights are calculated according to the $tf-idf$ schema, where the number of documents in which a given term occurs is calculated in comparison to all documents in the collection (i.e. on the basis of a search engine index).
3. Each term t_i , which belongs to all relevant documents, is assigned wz_i' weight, which is the minimum over weights d_i of term t_i in the relevant documents (the way of calculating the weight wz_i' was inspired by the research on grouping collection of documents made by Voorhees [19]). The term t_i is further considered as a potential significant term.
4. The df rule is next applied to above identified set of potential significant terms. Only these terms are considered for the further analysis which df value is between df_{min} value and df_{max} value (the values were set experimentally).
5. The *cue validity* factor cv_i is calculated for each term t_i selected at the step 4.
6. The weight $wz_i(1)$ is calculated for each t_i selected at the step 4.
7. The threshold τ , called *significance factor* ι is applied for further terms filtering and ranking. If the wz_i weight of the term t_i is higher than ι , we assume t_i to be a proper *significant term* tz_i .

Only terms occurring in all relevant documents can be included into a set of significant terms. The above criteria of terms selection have been aimed on finding only terms describing the domain of user interests and improving the number of relevant documents in the answer.

3 User Profile

The IR system is defined by four elements: set of documents D , user profiles P , set of queries Q and set of terms in dictionary T . There is retrieval function $\omega: Q \rightarrow 2^D$. Retrieval function returns the set of documents, which is the answer for query q . The set $T = \{t_1, t_2, \dots, t_n\}$ contains terms from documents, which have been indexed by Web retrieval system and is called dictionary. D_q is representing set of relevant documents among the documents retrieved D_q' for query q : $D_q' = \omega(q, D)$ and $D_q \subseteq D_q'$.

The *user profile* $p \in P$ is represented by set of pairs:

$$p = \{\langle s_1, sp_1 \rangle, \langle s_2, sp_2 \rangle, \dots, \langle s_l, sp_l \rangle\} \quad (2)$$

where: s_j – a user query pattern, sp_j – a user subprofile (user query pattern indicates one user subprofile univocally).

For profile p there is function π , which maps: user's query q , the set of retrieved relevant documents D_q and the previous user profile p_{m-1} , into a new user profile p_m . The function π determines the profile modifications. Thus, the profile is the following multi-attribute structure: $p_0 = \emptyset$, $p_m = \pi(q_m, D_q, p_{m-1})$. For the user profile we define also the set of *user subprofiles* SP (see below).

The user profile is created on the basis of *user's verification of the documents* retrieved by the system. During verification the user points out these documents which he considers relevant.

The *user query pattern* s_j is a Boolean statement, the same as the user query q : $s_j = r_1 \wedge r_2 \wedge r_3 \wedge \dots \wedge r_n$, where r_i is a term: $r_i = t_i$, a negated term: $r_i = \neg t_i$ or logical one: $r_i = 1$ (for terms which does not appear at the question). The user query pattern s_j indicates the subprofile and is connected to only one subprofile.

The *user subprofile* $sp \in SP$ is n -dimensional vector of weights of terms from the relevant documents: $sp_j^{(k)} = (w_{j,1}^{(k)}, w_{j,2}^{(k)}, w_{j,3}^{(k)}, \dots, w_{j,n}^{(k)})$, where SP is the set of subprofiles, n – the number of terms in dictionary T : $n = |T|$, $w_{j,i}^{(k)}$ – the weight of the significant term t_{z_i} in subprofile after the k -th subprofile modification.

The position of weight $w_{ji}^{(k)}$ in the subprofile (its co-ordinate in the vector of the subprofile) indicates the significant term $t_{z_i} \in T$. The terms from dictionary T are an indexing terms at Web search system, that index documents retrieved for the query q and these terms belong to these relevant documents.

The *weight of significant term* t_{z_i} in subprofile is calculated as following:

$$w_{j,i}^{(k)} = \frac{1}{k} ((k-1)w_{j,i}^{(k-1)} + wz_{i}^{(k)}) \tag{3}$$

where: k – the number of retrievals of documents made so far for this subprofile, i – the index of term in the dictionary T , j – the index of a subprofile, $w_{j,i}^{(k)}$ – the weight of the significant term t_{z_i} in the subprofile after the k -th modification of the subprofile¹, which is indicated by the pattern s_j (i.e. after the k -th document retrieval with the use of this subprofile), $wz_{i}^{(k)}$ – the weight of the significant term t_{z_i} in the k -th selection of these terms.

4 Modification of User Profile

The *adaptive user profile* expresses the translation between the terminology used by the user and the terminology accepted in some field of knowledge. This translation describes the meaning of the words used by the user in a context fixed by relevant documents and it is described by assigning to the user's query pattern s_j a subprofile ('translation') created during the process of significant terms t_{z_i} selection from relevant documents of an answer. We assume the following designations: q – the user query, D_q' – the set of documents retrieved for the user query q , $D_q' \subseteq D$, D_q – the set of documents pointed by the user as relevant documents among the documents retrieved for user query q , $D_q \subseteq D_q'$.

As it was described above, the user profile p_m is the representation of the user query q , the set of relevant documents D_q and the previous (former) user profile p_{m-1} . After every retrieval and verification of documents made by the user, the profile is modified. The modification is performed according to the following procedure:

¹ The weight, called *cue validity*, is calculated according to a frequency of term t_{z_i} in relevant documents retrieved by the system in k -th retrieval and a frequency of this term in whole documents of the collection.

$p_0 = \emptyset$, $p_m = \pi(q_m, D_q, p_{m-1})$ where p_0 – the initial profile, this profile is empty, p_m – the profile after m -times the user has asked different queries and after each retrieval the analysis of relevant documents was made.

Traditionally, a user profile is represented by one n -dimensional vector of terms describing user interests. User interests change, and so should the profile. Usually changes of a profile are achieved by modifications of weights of the terms in the vector. After appearance of queries from various domains, modifications made for this profile can lead to an unpredictable state of the profile. By the unpredictable state we mean a disproportional increase of the weights of some terms in the vector representing the profile that could not be connected with an increase of user interests in the domain represented by these terms. The weights of terms can grow, because of high frequency of these terms in the whole collection of documents, regardless of the domain of actual retrieval.

The representation of a profile as one vector could also cause ambiguity during the use of this profile for query modification. At certain moment a query refers only to one domain of user's interests. To use the profile mentioned above we need a mechanism of choosing from the vector of terms representing various users' interests only these terms that are related to a domain of current query. To obtain this information, usually knowledge about relationship between terms from a query and a profile, and between terms in profile is needed. In literature, this information is obtained from a co-occurrence matrix created for a collection of documents [15] or from a semantic net [9]. One of disadvantages of presented approaches is that two mentioned above structures, namely a user profile and a structure representing term dependencies, should be maintained and managed for each user and also that creating the structure representing term relationships is difficult for so diverging and frequently changing environment as the Internet.

There are no such problems for the user profile p created in this contribution. After singular retrieval, only weights of terms from the subprofile identified by pattern s_j (identical to users' query) are modified, not weights of all terms in the profile. Similarly, when the profile is used to modify user query, the direct translation between the current query q and the significant terms from the domain associated with the query is used. In the profile p , between a single user query pattern s_j and a single subprofile sp_j a kind of mapping exists that represents this translation.

In Web search system the user profile is created during a period of time – during sequence of retrievals. There could appear a problem how many subprofiles should be kept in the user profile. We have decided that only subprofiles that are frequently used for query modifications should not be deleted. If a subprofile is frequently used, it is important for representing users' interests.

The modification of the user subprofile sp is made always when from the set of relevant documents pointed out from retrieved documents by the user, some significant terms tz_i are determined. The $w_{j,i}^{(k)}$ weights are modified only for these terms and only in one appropriate subprofile identified by the user query pattern s_j . The term tz_i weight is calculated according to (3). During each retrieval cycle the modification takes place only in one subprofile and for all significant terms tz_i obtain during the k -th selection of these significant terms from the relevant documents retrieved for the

query q , which was asked k -th time. If the modification took place for significant terms t_{z_i} for every subprofile in whole user profile, it would cause disfigurement of significant terms importance for single question.

5 Application of User Profile

The user profile contains terms selected from relevant documents. These terms are good discriminators distinguishing relevant documents among the other documents of the collection and these terms represent the whole set of relevant documents.

The application of user profile p is performed during each retrieval for a user query q . One of the main problems is the selection of significant terms t_{z_i} for query modification. Not all significant terms from a subprofile will be appropriate to modify the next user's query, because the query becomes too long.

If the user asks a new query q_j to the Web search system, a new pattern s_j and a subprofile identified by this pattern are added to the profile. The subprofile is determined after the analysis of relevant documents. If user asks the next query q_k and this query is the same as the previous query q_j , the given query is modified on the basis of the user profile. The modified query is asked to the Web search system, retrieved documents are verified by the user and the subprofile in user profile is brought up to date. After each use of the same query as query q_j , the subprofile identified by the pattern s_j represents user's interests described at the beginning by the query q_j even better. Each retrieval, with the use of the subprofile identified by the pattern s_j , leads to query narrowing, a decrease in the number of answer documents, an increase in the number of relevant documents.

The user profile can be used for query modification if a pattern s_j existing in the profile is *identical* to the current query q_i or *similar* to the current query q_i . For example for the queries: $q_a = t_1 \wedge t_2 \wedge t_3 \wedge t_4$, $q_b = t_1 \wedge \neg t_2$, the patterns: $s_1 = t_1 \wedge t_2 \wedge t_3 \wedge t_4$, $s_2 = t_1 \wedge \neg t_2$ are identical to the queries q_a , q_b , respectively, and the patterns: $s_2 = t_2 \wedge t_4$, $s_4 = t_1 \wedge t_2$, $s_5 = t_1 \wedge t_3$, $s_6 = t_2$ are similar to the query q_a .

If a pattern s_j is identical to the current user query q_i , the current user query q_i is replaced by the best significant terms t_{z_i} from the subprofile identified by the pattern s_j . The weights of these terms are over $\tau_{profile}$ – a dynamically calculated threshold. If in the user profile there are a few patterns that are similar to the current user query q_i , all significant terms t_{z_i} from all subprofiles identified by these patterns are taken under consideration. The weights of all significant terms t_{z_i} from the subprofiles identified by the similar patterns are summed. The n -dimensional vector of $R=(r_1, r_2, \dots, r_n)$ is created. The ranking of all these significant terms is made and significant terms, whose weights are over $\tau_{profile}$ dynamic threshold, replace the current user query q_i .

6 Experiments

The *adaptive user profile*, called the *Profiler*, is implemented as a part of Web search system. The user profile is used as a mechanism of retrieval personalisation, i.e. by user query modifications. The modification of user query takes place as a result of user interaction with a search engine (i.e. a verification of documents). After verifica-

tion, the system automatically asks the modified query to the search engine and presents the new answer to the user.

The experiments were forked into two directions. The first aim was to establish all parameters (i.e. thresholds) for the *Profiler*. The second aim – to verify the usefulness of the multi-attribute adaptive profile, i.e. more relevant documents retrieved at every retrieval and less numerous answers.

In the test environment, testing sets of documents were established, where relevant documents were identified by the group of 13 persons - real users of Web search system. Three types of documents sets were used: *dense sets*, *loose* and *mixed sets*. The dense sets of relevant documents consist of documents describing one domain of user interests; the strongly similar documents were assessed by users. The loose sets of documents consist of not similar to each other documents from different domains of user interests. The mixed sets of documents consist of subsets of closely related documents, identifying one domain of interests, and a number of documents from different domains of user interests.

From dictionary T the number of 50 random queries were generated. Each random query was asked to the search engine. If in the answer there were relevant documents, the randomly generated query was modified – the significant terms replace the preliminary query. The modified query was automatically asked to the search engine and the next relevant documents were found from testing sets of documents.

Each stage of above described cyclic process is called *iteration*. Iterations were repeated until all relevant documents from the dense sets of relevant documents were found or no changes in number of relevant documents were observed (for the loose and mixed sets of relevant documents).

Table 1. Measures of retrievals improvement made during experiments

	Percent of modified queries			Effectiveness %DR			
	improvement in precision $Prec_m$	no improvement in precision $Prec_m$	partial improvement in precision $Prec_m$	75-100%	50-75%	25-50%	0-25%
dense sets	82 %	12 %	6 %	54 %	10 %	18 %	18 %
loose sets	67 %	12 %	21 %	0 %	3 %	68 %	29 %
mixed sets	58 %	18 %	24 %	6 %	36 %	29 %	29 %

For every random query in experiments: the number of all retrieved documents D'_q , effectiveness %DR (percent of relevant documents retrieved by the modified queries from the set of all relevant documents in the test collections), and precision $Prec_m$ (for the first $m=10, 20, 30$ documents in the answer) at every iteration were calculated. These were retrieval improvement measures for the proposed method.

The retrievals made during experiments for dense sets of relevant documents confirmed that for most of the modified queries the retrieval results were better in comparison to the preliminary query. For over 82% of preliminary queries, the $Prec_m$ and the %DR were increasing with every iteration of query modification (Table 1). The number of all retrieved documents diminishes with every iteration.

The retrievals made for loose sets of relevant documents showed that a method of adaptive profile creation, modification and application assure that all modified queries (from one starting query) always focus on the same field of user interests.

In experiments for loose sets of relevant documents, for more then 67% of preliminary queries, the above measures rose with each iteration of query modification. The number of all retrieved documents diminishes as well. For the rest of the questions at that part of experiments, the measured parameters were worse, because as the answer only single documents were found by each modified query. No similar documents were found because of the structure of the loose sets.

7 Conclusions and Future Work

The *adaptive user profile* is a new and universal approach to the representation of user's interests and preferences. The profile includes both interests given explicitly by the user, as a query, and also preferences expressed by the valuation of relevance of retrieved documents. The important task of this profile is to express field independent translation between terminology used by user and terminology accepted in some field of knowledge. This universal translation is supposed to describe the meaning of words used by user (in user query) in context fixed by the retrieved documents (i.e. user subprofile). The experiments confirmed that during retrieval process in Web search system the modified query become more precise and user receives valuable support during query formulation. The query is modified in a way that, during next retrievals user receives the set of retrieved documents that definitely meet his/her information needs. In the future, some experiments need to be done with a bigger group of WWW users, who will retrieve and assess the documents from the Web.

Acknowledgments. This contribution was partially supported by Polish Ministry of Science and Higher Education under grant no. N N519 407437.

References

1. Ambrosini, L., Cirillo, V., Micarelli, A.: A Hybrid Architecture for User-Adapted Information Filtering on the World Wide Web. In: Proc. of the 6th Int. Conf. on User Modeling, pp. 59–62. Springer, Heidelberg (1997)
2. Asnicar, F., Tasso, C.: ifWeb: a Prototype of User Model-Based Intelligent Agent for Document Filtering and Navigation in the World Wide Web. In: Proc. of the Workshop Adaptive Systems and User Modeling on the World Wide Web, UM 1997. Springer, Heidelberg (1997)
3. Billsus, D., Pazzani, M.: A Hybrid User Model for News Story Classification. In: Proc. of the 7th Int. Conf. on User Modeling, UM 1999, Banff, Canada, pp. 99–108. Springer, Heidelberg (1999)
4. Benaki, E., Karkaletsis, A., Spyropoulos, D.: User Modeling in WWW: the UMIE Prototype. In: Proc. of the 6th Int. Conf. on User Modeling, pp. 55–58. Springer, Heidelberg (1997)
5. Bhatia, S.J.: Selection of Search Terms Based on User Profile. Comm. of the ACM (1992)

6. Bull, S.: See Yourself Write: A Simple Student Model to Make Students Think. In: Proc. of the 6th Int. Conf. on User Modeling, pp. 315–326. Springer, Heidelberg (1997)
7. Collins, J.A., Greer, J.E., Kumar, V.S., McCalla, G.I., Meagher, P., Tkatch, R.: Inspectable User Models for Just-In Time Workplace Training. In: Proc. of the 6th Int. Conf. on User Modeling, pp. 327–338. Springer, Heidelberg (1997)
8. Daniłowicz, C.: Modelling of user preferences and needs in Boolean retrieval systems. *Information Processing and Management* 30(3), 363–378 (1994)
9. Davies, N.J., Weeks, R., Revett, M.C.: Information Agents for World Wide Web. In: Nwana, H.S., Azarmi, N. (eds.) *Software Agents and Soft Computing: Towards Enhancing Machine Intelligence*. LNCS(LNAI), vol. 1198, pp. 81–99. Springer, Heidelberg (1997)
10. Goldberg, J.L.: CDM: An Approach to Learning in Text Categorization. *International Journal on Artificial Intelligence Tools* 5(1 and 2), 229–253 (1996)
11. Indyka-Piasecka, A., Piasecki, M.: Adaptive Translation between User's Vocabulary and Internet Queries. In: Proc. of the IIS IPWM 2003, pp. 149–157. Springer, Heidelberg (2003)
12. Daniłowicz, C., Indyka-Piasecka, A.: Dynamic User Profiles Based on Boolean Formulas. In: Orchard, B., Yang, C., Ali, M. (eds.) *IEA/AIE 2004*. LNCS(LNAI), vol. 3029, pp. 779–787. Springer, Heidelberg (2004)
13. Jeapes, B.: Neural Intelligent Agents. *Online & CDROM Rev.* 20(5), 260–262 (1996)
14. Maglio, P.P., Barrett, R.: How to Build Modeling Agents to Support Web Searchers. In: Proc. of the 6th Int. Conf. on User Modeling, pp. 5–16. Springer, Heidelberg (1997)
15. Moukas, A., Zachatia, G.: Evolving a Multi-agent Information Filtering Solution in Amalthea. In: Proc. of the Conference on Agents, Agents 1997. ACM Press, New York (1997)
16. Qiu, Y.: Automatic Query Expansion Based on a Similarity Thesaurus. PhD. Thesis (1996)
17. Salton, G., Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management* 24(5), 513–523 (1988)
18. Seo, Y.W., Zhang, B.T.: A Reinforcement Learning Agent for Personalised Information Filtering. In: Int. Conf. on the Intelligent User Interfaces, pp. 248–251. ACM, New York (2000)
19. Voorhees, E.M.: Implementing Agglomerative Hierarchic Clustering Algorithms for Use in Document Retrieval. *Inf. Processing & Management* 22(6), 465–476 (1986)

A Method for Web-Based User Interface Recommendation Using Collective Knowledge and Multi-attribute Structures

Michał Malski

Institute of Computer Science, The State Higher Vocational School in Nysa,
ArmiiKrajowej 7 Street, 48-300 Nysa, Poland
malskim@pwsz.nysa.pl

Abstract. This paper presents a framework of a method for the problem of web-based user interface personalization and recommendation using collective knowledge (coming from a collection of existing users) and multi-attribute and multi-value structures. In this method a user profile consists of user data and system usage path. For a new user the usage path is determined basing on the paths used by previous users (the collective knowledge). This approach involving the recommendation methods may be applied in many systems which require such mechanisms. The structure of user profile and an algorithm for recommendation are presented.

Keywords: user interface recommendation, recommendation system, usage recommendation, interface personalization.

1 Introduction

Interactiveness of modern websites is achieved among other things by the adjustment of an interface to a user profile. It refers not only to the aesthetics, but also the contents of the interface. Recommendation mechanisms have been applied and valued in many different fields such as [10,14]: content filtering on news websites (Internet news and information services), personalized e-news papers, online shopping, e-learning systems as well as movies, documents, music or travel recommendation.

As stated in numerous articles about recommendation, one of its most important elements is a user model. It usually contains [2,13]: data about a user (user data) and data about a system used by a user (usage data). Data about a user is most frequently obtained by filling-in fields survey during the registration process. Such data make it possible to create an initial user profile. During an interaction between a user and a system, the information about the way the system is used is being gathered. The user profile is modified in accordance with the acquired data, which renders recommendation more effective.

As far as news websites are concerned, it is important not only to filter the information based on a saved user profile, but also to locate such information in personalized user interface of such website. Contemporary web search engines adjust search

results to the information stored about the user, such as previous search activity, system use history or localization, which can be determined by IP address of a computer used to connect with search website.

Expanded and powerful information systems (engines) are behind interfaces which often look very simple. They are responsible for the interface adaptation and operate on the basis of activity of all the system users. Recommendation mechanisms operating in such systems are usually created for a particular target and they work best only there. Use of particular mechanism for another target often requires fundamental modifications, algorithm changes or recreating everything altogether.

The modern Internet user becomes accustomed to the many features of websites such as: the location of elements on the page, sound events, sequence of events, background, coloring or manner of exploration information. This article describes the problem of Web-based interface recommendation for a new user based on accumulated knowledge on how the system interface is used by other users. Profile of the new system user is created on the basis of data collected during the registration process. Based on the profile new user is qualified to the group of existing users who already have their own system usage path (their own interface). The concept presented below is interface recommendation for a new user, which is the best representative of the interfaces of all existing members of the group. In this work, as in many others on this subject [13, 16], it was suggested a description of the interface by states. The recommendation problem was reduced to the selection sequence of states from initial state to the final state of the interface by selecting the most frequented passage transitions. The proposed solutions take into account the fact, and the sequence of states in the recommended interface according to the map transitions between states. In the following described concept, the structure of a single state (state parameters and their values) affect on determination of the interface for the new user. Proposed function of distance between two system usage path to determine the closest path to the other, have been defined on the basis of a function of distance between two states of the interface. This innovation seeks to increase the effectiveness of the above problem solution and therefore the probability of misguided recommendation.

Movie recommendation system as well as book selling system will not work in the same way as mechanisms designed to recommend items in an on-line home appliances shop. Such systems differ at the fundamental assumption level. After an order has been made, the film recommendation system is meant to take into account the user's preferences and create a list of items similar to those from the previous orders. On the fundamental level it is known that that the system should provide a list of films of the same genre, same directory, with the same actors or even take into account items ordered by other customers of similar customer profile or similar shopping history. Recommendation of items in an on-line home appliances shop should not recommend products from the same category as previous ordered product.

The next part, there is a description of a complex adaptive user interface system with the use of recommendation methods and related problems. Section 3 presents a concept of possible solution for the problem mentioned earlier. Section 4 is an overview of concepts of user's interface personalization systems and recommendation systems which were suggested in other articles. Conclusions and the elements of future works are presented in the last section.

2 User Interface Recommendation System

During the user registration process, it is possible to obtain user's data on the basis of the fill-in fields on a registration webpage and to create a user's profile. Based on the created profile, the system classifies the user to the group of users with similar profiles.

The information about the use of the system interface by each user from a given group is stored by the engine working in the background of the website. To describe the interface usage path, it is needed to specify interface state in the following way:

$$s_i = (ID_i, A_i, V_i),$$

where:

ID_i - the identifier of interface state, $i \in \{1, \dots, N\}$,

N - the number of elements of the set of all possible interface states,

$A_i \subseteq A$ - the set of attributes describing interface state with number i ,

A - the set of all attributes,

$V_i \subseteq V$ - the set of values of attributes from A_i set for state with number i ,

V - the set of all possible values of attributes.

It should be emphasized that $s_i \in S$, where $S = \{s_1, \dots, s_N\}$ is a set of all possible interface states. It is very important to create a map of interface by designating interface states, into which they can be switched from their actual state. Let $R \subseteq S \times S$ be the relation containing pairs of states such as (s_i, s_j) if it is possible to switch the interface from state s_i to state s_j , which will be identified as $(s_i, s_j) \in R$, where $i, j \in \{1, \dots, N\}$. On the basis of such a defined relation, it is possible to define the system usage path, which will describe the scenario of system interface use by a particular user. For a given initial state and a given final state, the scenario of system use will be a path of states which can switch the interface from the initial to the final state according to relation R .

In paper [13] the author suggests creating oriented acyclic graph to illustrate the rules mentioned above. Such graph has the start node, finish node, and each middle node is a pair of interface attribute and its value. The problem of interface recommendation is defined as finding best suiting path from start to finish node. An example of user interface graph is presented below in Figure 1.

Personalization of user interface basing on recommendation methods is a task relying on finding scenario of system use for a new user classified to the group. Such scenario should be determined by the previous group member's sceneries. The author of the [13, 15], used the ant colony metaphor to solve the recommendation problem. According to this idea, the proposition of the interface for a new user should be generated by the path from the start to the finish node of the graph, which was most frequently chosen by the ant agent (a member of ant colony).

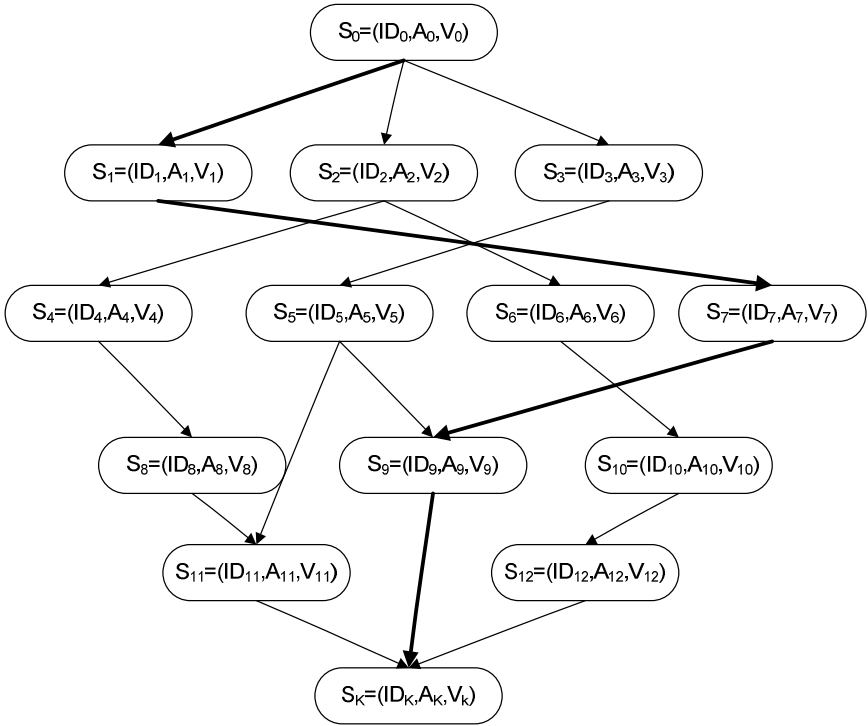


Fig. 1. Sample user interface graph

On the basis of the above graph, it is possible to generate the set of all possible scenarios of system interface use. Let $up_i = (up_i^{(1)}, \dots, up_i^{(n)})$ be a possible usage path, which follows conditions pointed below:

$up_i^{(1)} = s_0$ indicates the initial interface state,

$up_i^{(n)} = s_K$ indicates the finish interface state,

$up_i^{(j)} = s_l, l \in \{1, \dots, N\}$, indicates the state, for which $(up_i^{(j)}, up_i^{(j+1)}) \in R$ for each $j \in \{1, \dots, n-1\}$,

$i \in \{1, \dots, M\}$, where M is a cardinal number of UP set as a set of all possible scenarios of system interface use, which is created on basis of relation R ,

$up_i \in UP$.

As can be seen, scenarios can have different lengths, and therefore not the shortest path should be proposed by a recommendation task, but the most suitable to usage paths of all group members, where a new user is classified. For a given subset of UP set, the task relies on finding system a usage path, which minimizes the sum of the distance function between this path and each path from a given subset.

The previous studies [5, 6, 7] concern the problem of recommendation, in which usage paths are constant length sequences of numbers from the same set. Recommendation task relies on appointing the sequence which minimizes the sum of the distances between this sequence and all another form set. This sequence should be proposed for a new member of the group. In the mentioned papers, for the described problem, the distance function was proposed based on [4, 11, 12]. There can be a proposed algorithm for solving minimization sum of the distances problem and its statistical verification. For the problem described in this paper, usage paths do not have constant length and elements of have not to be numerous. The elements of the path are states and have their own structure and paths include different states. Therefore, it is necessary to create a new distance function for the problem described here.

For the interface state defined above $s_i = (ID_i, A_i, V_i)$ the set A_i is included in A set as in set of all possible attributes used to describe all interface states?. It is possible to define the function which determines values of attributes for each element from A set used to create state with index i :

$$r_i : A \rightarrow \{V_i, null\}$$

where $r_i(a)$ is an element of V_i or $null$, if $a \notin A_i$.

In this way, the tuple is created, which for a state with number i will be as follows:

a_1	a_2	a_3	a_4	...	$a_{card(A)}$
$r_i(a_1)$	$r_i(a_2)$	$r_i(a_3)$	$r_i(a_4)$...	$r_i(a_{card(A)})$

and for state with number j :

a_1	a_2	a_3	a_4	...	$a_{card(A)}$
$r_j(a_1)$	$r_j(a_2)$	$r_j(a_3)$	$r_j(a_4)$...	$r_j(a_{card(A)})$

The distance between such two tuples, and hence the distance between two states can be defined in the following way:

$$d(s_i, s_j) = \sum_{k=1}^T \frac{\mu_k(s_i, s_j)}{T},$$

where:

T is a cardinal number of A set,

μ_k is function created on the basis of function ρ described in [11,12], which is based on determining the value of participation of elements of given set in the distance between two subsets of this set. μ_k is defined as below:

$$\mu_k(s_i, s_j) = \begin{cases} 0 & \text{for } r_i(a_k) = r_j(a_k) \\ \alpha_k & \text{for } r_i(a_k) \neq r_j(a_k) \text{ and } r_i(a_k), r_j(a_k) \text{ not null,} \\ 1 & \text{for } r_i(a_k) = null \text{ xor } r_j(a_k) = null \end{cases}$$

where $0 < \alpha_k < 1$ is a constant number sets for each one of attributes.

Now, after defining the distance function between two states, a distance function between two usage paths can be defined. For the given state $s \in S$ and given usage path $up \in UP$, it is possible to determine an element of up , for which the distance to the state s is minimal. This optimal element will be denoted by up' and determined in the following way:

$$d(s, up') = \min_{j=1, \dots, l} d(s, up^{(j)}), \text{ where } l \text{ is number of element of } up \text{ path.}$$

For the given two usage paths $up_1, up_2 \in UP$ the distance between element of up_1 path with number i and optimal state from up_2 path (up'_2), will be calculate as below:

$$DU(up_1^i, up_2) = d(up_1^i, up'_2) \cdot \frac{|pos(up_1^i, up_1) - pos(up'_2, up_2)|}{k + l},$$

where:

k - the number of elements of up_1 ,

l - the number of elements of up_2 ,

pos function returns the number of states in the path.

Now it is possible to define the distance function between the two given paths in the following way:

$$D(up_1, up_2) = \frac{\sum_{i=1}^k DU(up_1^i, up_2) + \sum_{j=1}^l DU(up_2^j, up_1)}{2}.$$

Based on the above defined distance function, a formal condition for the optimal system usage path up^* can be presented. For such set, which size is equal n the condition is as follows:

$$\sum_{i=1}^n D(up^*, up_i) = \min_{up \in UP} \sum_{i=1}^n D(up, up_i).$$

Path up^* that fulfills the above condition is a solution of the recommendation problem, and should be proposed for the new member of the group of users. It also translates into the appearance of interface offered for the user. The path up^* can also be modified after it is determined in order to further reduce the sum of distances to other members' paths. The above concept is presented in the next section.

3 Algorithms for Recommendation

For a given set $UP_G \subseteq UP$, which is a set of system usage paths belonging to a particular group of users, by $up_{Gi} \in UP_G$ we will denote the usage path of group G member with

number i . Assuming the number of users equal to n ($card(UP_G) = n$), the task described above consist in determining the usage path for a new user with the number $n+1$, which fulfills the criterion defined in the preceding paragraph. To accomplish this, we determine the path from the set UP_G , for which the sum of distances to the other paths from this set is the lowest, according to the following procedure:

Input: UP_G - the set of system usage paths of users, which are members of a given group.

Output: Path $up_{G\min} \in UP_G$ that fulfills the following condition:

$$\sum_{i=1}^n D(up_{G\min}, up_{Gi}) = \min_{up \in UP_G} \sum_{i=1}^n D(up, up_{Gi})$$

BEGIN

1. $up_{G\min} = up_{G1}$ and minimal sum of distances $SoD_{\min} = \sum_{i=1}^n D(up_{G1}, up_{Gi})$

2. For $j=2$ to n do

If $SoD_{\min} > \sum_{i=1}^n D(up_{Gj}, up_{Gi})$ then

$$up_{G\min} = up_{Gj} \text{ and } SoD_{\min} = \sum_{i=1}^n D(up_{Gj}, up_{Gi})$$

END.

In this way, we can get the path belonging to the set UP_G , which minimizes the sum of the distances to all other paths belonging to this set. However, this does not mean that in the UP - which is the set of all possible paths in the described system, there is no path whose sum of distances to all the paths of the set UP_G is smaller than for the path $up_{G\min}$ designated above.

Search of the whole set UP , which is in fact finite, may take too much time and be an impossible task after number of elements of this set has increased beyond a certain number. As a solution to this recommendation problem, we can assume $up_{G\min}$ path, which can be further modified to reduce the sum of distance function to all the usage paths owned by users from the same group. Such an approach does not guarantee that the resulting solution will be optimal.

An example of such modification is shown below:

Input: $up_{G\min}$, UP_G , S and l as length of $up_{G\min}$

Output: Path $up_{G\min}^*$ that fulfills the following condition:

$$\sum_{i=1}^n D(up_{G\min}^*, up_{Gi}) \leq \sum_{i=1}^n D(up_{G\min}, up_{Gi})$$

BEGIN

$$1. \quad up_{G \min}^* = up_{G \min}, \quad SoD_{\min}^* = \sum_{i=1}^n D(up_{G \min}, up_{Gi})$$

2. For $j=2$ to $l-1$ do

For each $s_x \in S$ for which $(up_{G \min}^{*(j-1)}, s_x) \in R$ and $(s_x, up_{G \min}^{*(j+1)}) \in R$

BEGIN

modified path *temp* is equal $up_{G \min}^*$ but $up_{G \min}^{*(j)}$ is replaced by s_x

If $SoD_{\min}^* > \sum_{i=1}^n D(temp, up_{Gi})$ then

$$up_{G \min}^* = temp \text{ and } SoD_{\min}^* = \sum_{i=1}^n D(temp, up_{Gi})$$

END;

END.

This modification involves exchanging individual elements of the path with their counterparts in order to comply with the graph and thus the whole system. Any such established path is an element of the set UP . After each of these substitutions, the sum of functions is checked to the paths of all users is checked, which ultimately determines the substitution of an item on a permanent one or a rejection of such change.

When the UP_G is a large collection and its size will cause problems, it is desirable to divide this collection into subsets and to use this solution for different subsets. Solutions for a subsets, obtained in this way, will create concentration, for which the re-application of the solutions will appoint centroid, which will be the solution for the whole set UP_G .

4 Related Works

In many works about adaptive user interfaces, we can see the different methods of acquiring knowledge about the users and using the system. Job [1] describes a very useful method of gaining knowledge by analyzing web server logs. First, the author suggests cleaning server access log files by deleting unneeded downloaded files other than HTML documents such as the style files, graphic files, music files and references to non-existent pages, etc. So pre-filtered logs should be subjected to a data mining process to link entries to each user and in general create individual user profiles. The described method of acquiring knowledge is particularly useful in the system of recommendations which is being created, where their own methods of data collection have not been implemented yet. The method is connected with many problems concerning the interpretation of the logs associated with the proxy servers, a lot of queries from a single IP address in a short period of time, which could indicate a network hidden behind a single external address by using NAT.

A similar approach to the problem of data collection is presented in jobs [8, 9], where data are also collected based on web server logs and then analyzed using WUM techniques (Web Usage Mining).

For these problems, we can create a graph-oriented as in the aforementioned work [13]. Nodes of this graph would be URLs, whereas links between them would mean a move from one site to another. Based on the logs, there is a possibility of checking how often the path between two nodes was visited. Recommended use of the path of the system (transitions between states of the interface) would be composed of the most popular crossing between the nodes of the graph.

Adaptive computerized method of examining for an intelligent learning system itself is presented in the article [3]. This system initially adapts itself on the basis of data we collect about a user during the registration process. On the basis of these data a parameter is assigned and it determines the choice of the first question. It aims to determine the query that most closely matches his preferences. Generally, the choice of the next queries is influenced by the already asked queries. After each answer, the system updates the user's fitness level (proficiency level) and estimates the level of knowledge. Everything is based on the model of the IRT (Item Response Theory), and the calculation of the function IRF value (Item Response Function) and IIF (Item Information Function).

5 Conclusions

Adapting a user interface is one of many problems that can be solved by the use of systems recommendations. Above, there has been shown the concept of a complex recommendation system which uses the methods of finding consensus and its application in the adaptation of the user interface. However, the system, which describes the concept, may be used wherever it is desirable to use the recommendations.

There is a criterion defined which is to be fulfilled by the solution of the recommendation's problem and there is also distance function defined which is needed to determine the criterion. The problem was also presented with a graph and it illustrates the path of a solution from start to end node which is most similar to the paths of all users of the group. This does not necessarily mean the shortest path in this graph.

The general concept of solving the recommendation problem fulfilling the described assumptions has been presented. The detailed algorithm and its implementation, as well as the results of statistical verification will be aspects of the future research.

Acknowledgements. This paper was partially supported by Polish Ministry of Science and Higher Education under grant no. N N519 407437 (2009-2012).

References

1. Ahmad, A.M., Hijazi, M.H.A.: Web Page Recommendation Model for Web Personalization. In: Negoita, M.G., Howlett, R.J., Jain, L.C. (eds.) KES 2004. LNCS(LNAI), vol. 3214, pp. 587–593. Springer, Heidelberg (2004)

2. Kobsa, A., Koenemann, J., Pohl, W.: Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships. *Knowledge Eng. Rev.* 16(2), 111–155 (2001)
3. Koziarkiewicz-Hetmańska, A., Nguyen, N.T.: A Computer Adaptive Testing Method for Intelligent Tutoring Systems. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) *KES 2010. LNCS(LNAI)*, vol. 6276, pp. 281–289. Springer, Heidelberg (2010)
4. Kukla, E., Nguyen, N.T., Sobecki, J., Danilowicz, C., Lenar, M.: Determination of Learning Scenarios in Intelligent Web-Based Learning Environment. In: Orchard, B., Yang, C., Ali, M. (eds.) *IEA/AIE 2004. LNCS (LNAI)*, vol. 3029, pp. 759–768. Springer, Heidelberg (2004)
5. Malski, M.: An Algorithm for Inconsistency Resolution in Recommendation Systems and Its Application in Multi-Agent Systems. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) *KES-AMSTA 2009. LNCS(LNAI)*, vol. 5559, pp. 356–366. Springer, Heidelberg (2009)
6. Malski, M.: An Algorithm for Inconsistency Resolving in Recommendation Web-based Systems. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) *KES 2006. LNCS(LNAI)*, vol. 4252, pp. 251–258. Springer, Heidelberg (2006)
7. Malski, M.: Resolving inconsistencies in recommendation Web-based systems. In: *Proceedings of the 11th System Modelling Control Conference (SMC 2005)*, pp. 189–194 (2005)
8. Mican, D., Tomai, N.: Association-Rules-Based Recommender System for Personalization in Adaptive Web-Based Applications. In: Daniel, F., Facca, F.M. (eds.) *ICWE 2010. LNCS*, vol. 6385, pp. 85–90. Springer, Heidelberg (2010)
9. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization. *Data Mining Knowledge Discovery*, 61–82 (2002)
10. Montaner, M., Lopez, B., De La Rosa, J.L.: A Taxonomy of Recommender Agents on the Internet. *Artificial Intelligence Review* 19, 285–330 (2003)
11. Nguyen, N.T.: Methods for resolving conflicts in distributed systems. Monograph. Wrocław University of Technology Press (2002)
12. Nguyen, N.T.: Consensus System for Solving Conflicts in Distributed Systems. *Journal of Information Sciences* 147, 91–122 (2002)
13. Sobecki, J.: Ant Colony Metaphor Applied in User Interface Recommendation. *New Generation Computing* 26(3), 277–293 (2007)
14. Sobecki, J.: Hybrid Adaptation of Web-Based Systems User Interfaces. In: Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) *ICCS 2004. LNCS*, vol. 3038, pp. 505–512. Springer, Heidelberg (2004)
15. Sobecki, J., Tomczak, J.M.: Student Courses Recommendation Using Ant Colony Optimization. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) *Intelligent Information and Database Systems. LNCS*, vol. 5991, pp. 124–133. Springer, Heidelberg (2010)
16. Sobecki, J., Szczepański, L.: Wiki-News Interface Agent Based on AIS Methods. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) *KES-AMSTA 2007. LNCS(LNAI)*, vol. 4496, pp. 258–266. Springer, Heidelberg (2007)

Opinion Analysis from the Social Web Contributions

Kristína Machová

Dept. of Cybernetics and Artificial Intelligence, Technical University, Letná 9,
042 00, Košice, Slovakia

kristina.machova@tuke.sk

Abstract. The paper focuses on the automatic opinion analysis related to web discussions. It introduces a method for solving basic problems of opinion analysis (determination of word subjectivity, polarity as well as intensity of this polarity). The method solves the reversion of polarity by negation as well as determination of polarity intensity of word combinations. A dynamic coefficient for the word combinations processing is introduced and an implementation of the method is presented. In addition, the paper describes test results of the presented implementation and discussion of these results as well.

Keywords: opinion analysis, opinion classification, social web, sentiment analysis, web discussions, dynamic coefficient.

1 Introduction

Opinion analysis represents a domain, which is a firm part of the field of social web analysis. The social web can be considered as an upgrade of the classic web. The classic web can be illustrated with an idea of a world-wide billboard - anybody can publish some information piece or make it accessible for public inspection on the billboard (anybody who has necessary skills in web page creation - but considerably greater amount of web users have abilities only for reading this published information). On the other hand, the social web or web 2.0 reinforces social interactions among user and provides an opportunity for great majority of web users to contribute to web content. It can be said, that it increases the number of web content providers. Social interactions among users are enabled by communication within social nets, by the possibility to contribute to web discussions, and so on.

Specifically, discussion forums are large-scale data bases of opinions, attitudes and feelings of web users, who use the web for communication. Unlike classic data bases, they do not contain data in a structured form. For this reason, they need special methods for processing. One of such special methods is also opinion analysis. The main objective of opinion analysis is to summarize attitude of particular subscribers to some particular theme. This theme can be, for example, an evaluation of some product, political situation, person (e.g. active in politics), event or company.

Opinion analysis or opinion classification can be used in those fields where the aggregation of a large amount of opinions into integrated information is needed. The input to opinion classification can be represented by a large amount of discussion contributions (e.g. content of a discussion forum) and the output of the classification

is summarising information, for example “Users are satisfied with this product” or “People perceive this reform negatively”. From the point of view of a consumer, two kinds of information are important for decision making about purchase of a product. First, it is information about price and properties of the product, which usually are available on web pages of a producer or a seller. Second, it is information about satisfaction of other consumers with the product. The opinion classification can offer this information to prospective consumer. From the point of view of a producer, information about satisfaction and needs of consumers is also very important. The classic way of obtaining this information is performing market research. The market research carried out by telephone or by questionnaires is usually rather expensive and time consuming. The promptness of such information elicitation is a matter of principle. User contribution analysis provided by a system utilising opinion classification can offer the information about clients’ satisfaction more quickly.

2 Related Works

Sometimes, the introduced opinion analysis is denoted as opinion mining, because it focuses on the extraction of positive or negative attitude of a participant to commented objects with the aid of mining techniques applied to text documents. Opinion mining can be extended from the level of whole texts perception to the level of extraction of properties of those objects which match users’ interests [3]. Parallel approach to opinion mining is sentiment analysis [8]. Different access to web discussion processing is represented by the estimation of authority degree of some information sources, for example of actors contributing to discussion forums or social nets. An important technique for authoritative actors searching is visualization approach, which is introduced in [4]. Some effort was spent on semantically enrich algorithms for analysis of web discussion contributions [6].

Nowadays, opinion analysis has become an important part of social networks analysis. Existing opinion analysis systems use large vocabularies for opinion classification into positive or negative answer categories. Such approach was used in [2]. Authors studied accuracy of the opinion analysis of Spanish documents originated in the field of economic. This approach uses a regression model for classification into negative or positive opinions. Authors studied how quality depends on the granularity of opinions and rules, which were used in the regression model. Another study [5] was focused on the possibility of using lesser granularity without any significant precision decrease. The results of this study show no substantial difference between one and two parameter regression models as well as no statistically significant difference between models with different granularity. Thus, for example, simpler models can be used with the used sentiment scale reduced to five degrees only.

The presented approach uses a scale with five degrees for opinion classification as well, but it differs from the previous approaches in vocabulary cardinality. Our work focuses on creating vocabularies with strong orientation on the discussion domain, not so large but created directly from live discussions. We do not use regression models. First, words from discussions are classified into predefined categories and after that, this classification is transformed into another one enabling classification of the whole contribution into one of five degrees (strong negative, negative, neutral, positive and strong positive).

3 Basic Problems of Opinion Analysis

Three basic problems of opinion analysis are: *word subjectivity identification*, *word polarity (orientation) determination* and *determination of intensity of the polarity*. Opinion analysis focuses on those words, which are able to express *subjectivity* very well - mainly adjectives (e.g. ‘perfect’) and adverbs (e.g. ‘beautifully’) are considered. On the other hand, other word classes must be considered as well in order to achieve satisfactory precision, for example nouns (e.g. ‘bomb’) or verbs (e.g. ‘devastate’). The words with subjectivity are important for opinion analysis; therefore they are identified and inserted into the vocabulary. Words with subjectivity are inserted into the constructed vocabulary together with their polarity.

The *polarity of words* forms a basis for the polarity determination of the whole discussion. There are three basic degrees of polarity being distinguished: positive (e.g. ‘perfect’, ‘attract’), negative (e.g. ‘junk’, ‘shocking’, ‘absurdity’, ‘destroyed’) and neutral (e.g. ‘averaged’, ‘effectively’). This scale can be refined to use more possible levels if needed. The determination of the polarity of words is connected with a problem of word polarity reversion – the reversion can be done by using negation, for example ‘It was not very attractive film’. This problem serves as an argument for the extension of single words polarity determination to polarity determination of word combinations (considering whole sentences or parts of sentences).

The *intensity of word polarity* represents a measure of the ability of words to support the proof or disproof of a certain opinion. The polarity intensity of words can be determined according to a defined scale, which helps to classify words into more categories. Table 1 illustrates three such scales with different numbers of degrees.

Table 1. Scales using verbal or numerical representation of the intensity of word polarity

Number of Degrees	Scales of polarity intensity	
2	negative	Positive
6	weak, gently, strong negative	weak, gently, strong positive
8	-1, -2, -3, -4	1, 2, 3, 4

The polarity intensity can be expressed both verbally as well as numerically. The numerical representation is more suitable for subsequent processing by computers. Discussion contributions very often contain some word combinations, which increase (decrease) the weak (strong) intensity of polarity of an original word, for example: ‘surprisingly nice’, ‘high quality’, ‘markedly weaker’ and ‘extremely low-class’.

3.1 Classification Vocabulary Creation

In order to support the process of opinion analysis, it is necessary to create a vocabulary. The opinion analysis systems commonly utilise large vocabularies, which are called seed-lists. For example WorldNet can be used as a basis for the creation of such seed-list vocabulary. In accordance with [1], it is possible to derive tagsonomies from crowd. Similarly, we attempted to derive a vocabulary directly from web discussions. This vocabulary is specialized for a particular domain, the utilised web

discussions focus on. Since it is possible to use this vocabulary for classification of words into predefined categories, we denote it as a classification vocabulary.

Many of web discussion respondents do use literary language far from perfectly. Therefore our system of opinion classification has to be able of the adaptation to colloquial language of users of the Internet including slang, absence of diacritical marks, and frequent mistakes.

4 Design of Opinion Classification Method

The design of an opinion classification method has to consider all steps of the classification process and provide them in the right and logical sequence. The method we have designed solves the following problems:

- Basic problems of opinion analysis
- Word polarity reversion by negation
- Determination of the intensity of polarity
- Establishment of a dynamic coefficient
- Polarity determination of word combinations

Our access takes into account not only polarity of single words but also the intensity of polarity of word combinations including negation. Our method analyzes texts of discussion contributions from a certain domain and for this domain a classification vocabulary is generated from the given texts. The quality of the vocabulary and its cardinality play the key role in the process of opinion classification.

The method transforms textual content of a discussion contribution into an array of words. Each word with subjectivity is assigned a numerical value (numerical code) as it is illustrated in Fig. 1. This value represents the category of word polarity to which the given word belongs (see Table 2). Particular sentences are identified. First non zero value of word category starts the creation of word combination procedure. The length of a certain combination is limited by a coefficient K . Each combination of words is also assigned a numerical value which represents a polarity degree from the $\langle -3, 3 \rangle$ interval. The polarity degrees of all word combinations within the given text form the polarity of this text as a whole. Subsequently, the polarity of the whole discussion can be calculated from the polarities of all contributions (texts).

The whole contribution or discussion is considered to be positive/negative when it contains more positive/negative word combinations or contributions. The neutral contribution (discussion) contains the same number of positive and negative word combinations (contributions). This approach to neutrality determination is rather strict. A more benevolent approach uses the following rule for neutrality detection:

$$IF |Number_pozit - Number_negat| \leq H THEN neutrality$$

where threshold H represents range of neutrality, which can be changed by setting another value of the H parameter ($H \geq 1$ and it is an integer). Strict approach to neutrality with $H=0$ is more suitable for very short contributions, because such short contributions can contain only one sentence and only one positive or negative word. Wider neutrality range could absorb this word and subsequently the system of opinion classification can evaluate it as a neutral contribution. The wider neutrality range is more suitable for longer contributions processing.

4.1 Basic Problems Solution

In our approach, words with subjectivity are selected and the category value from a given scale (the scale from 0 to 9 is used) is assigned to each of these words, what is illustrated in Fig. 1. The words with positive polarity are classified to categories 1 or 8 (see Table 2). Similarly, the words representing negative polarity are classified to categories 2 or 9 and words with neutral polarity to 0 category.

Table 2. Categories of words polarity

weak positive and strong positive	1 and 8
weak negative and strong negative	2 and 9
Neutral	0
negation – polarity reversion	3
increasing of polarity intensity	4

To illustrate usage of these categories, Fig. 1 illustrates categorization of words into polarity categories based on the example ‘This mobile is marvellous and its functioning is reliable’. This word classification system also solves determination of the intensity of the polarity, because values 1 and 2 represent weak polarity in contrast to values 8 and 9, which represent strong polarity (being positive or negative). Thus, the designed method uses a five degree scale of the intensity of polarity determination (including neutral).

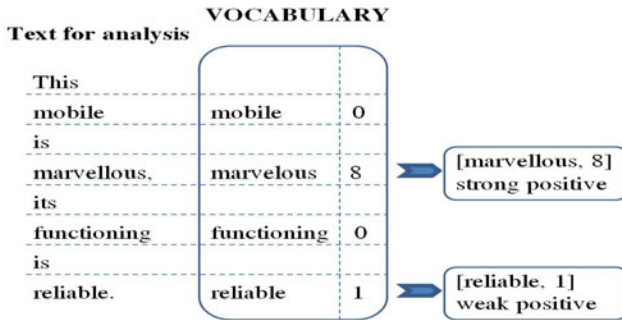


Fig. 1. Polarity determination of words from the sentence ‘This mobile is marvellous, its functioning is reliable’

There is one more addition in our design for determining the intensity of polarity. It is the category 4 used for each word, which increases the intensity of polarity of another word in the same word combination (e.g. ‘high quality’).

To summarise, the used polarity categories are introduced in Table 2. All words with subjectivity are expected to be inserted into the classification vocabulary together with their category codes.

4.2 Word Polarity Reversion by Negation

The reversion of word polarity caused by the usage of negation enables to reflect actual meaning and therefore to increase precision of opinion classification. The words, which represent negation (e.g. ‘none’, ‘no’) belong to the category 3. This category can be used only in the combination with another category (1, 2, 8 or 9). It changes positive polarity into negative polarity and vice versa within the same degree of intensity (weak or strong) as it can be seen in Table 3.

Table 3. Word polarity reversion by negation

3 + 1	3 + 8	3 + 2	3 + 9
negation + weak positive =	negation + strong positive =	negation + weak negative =	negation + strong negative =
<i>weak negative</i>	<i>strong negative</i>	<i>weak positive</i>	<i>strong positive</i>

The polarity reversion is a rather complicated issue due to the fact, that the structure of various sentences is not homogenous. For example, the sentence ‘This mobile isn’t reliable’ can be represented by the code 0031 (the code of a sentence is created by replacing each word with the number indicating its category). Another sentence ‘It isn’t, according to my opinion, reliable mobile’ has the same meaning but different code 03000010. The aim of our technique is to recognise various codes 0031 and 03000010 as opinions with the same polarity. Thus, there is a need of some dynamic coefficient, which enables to estimate an appropriate length of those word combinations, which will be processed together as one lexical unit. In general, it enables to process one sentence as two different combinations – lexical units.

4.3 Determination of the Intensity of Polarity

Words, which increase the intensity of polarity, have no polarity and their influence on polarity of a lexical unit can be evaluated only within a combination with the given lexical unit. These words belong to the category 4. Table 4 presents two different examples of such combinations.

Table 4. Analysis of lexical units with word increasing polarity intensity

This 0-neutral	mobile 0-neutral	is 0-neutral	totally 4 + intensity	conforming 1-weak positive
It 0-neutral	really 4 + intensity	drives 0-neutral	me 0-neutral	mad 2-weak negative

Both these combinations contain a word increasing the intensity of polarity. The word combinations are represented with codes 00041 and 04002. Words from the category 4 are usually adverbs (e.g. ‘very’, ‘really’, ‘totally’). Processing of the words enabling to increase the intensity of word polarity needs to use the dynamic coefficient in a similar manner as the negation processing.

4.4 Dynamic Word Combination Length

The designed method of opinion classification has an ambition to manage the variability of sentence structures using the dynamic coefficient K . The value of this parameter is being dynamically changed during processing of different lexical units. The dynamic coefficient adapts itself to the code length of a lexical unit (sequence of words) under investigation. The value K represents the number of words, which are included into the same word combination (beginning from the first non-zero word code in the sequence of words). In the case, when the value is higher than the number of words in the sentence, this value is dynamically decreased in order to ensure, that the combination contains only words from the investigated sentence, not from the beginning of the following sentence. A word combination can be shortened also in some other cases. For example, let us take the case $K=4$ while the combination 3011 is being processed. In this case, two disjunctive combinations are created 301 ($K=3$) and 1 ($K=1$). On the other hand, the value can be increased in some cases. Table 5 illustrates the principle of using the dynamical coefficient.

Table 5. Principle of employing the dynamical coefficient K (Words processed within one combination are given in bold.)

K	Never	buy	this	nice	mobile
1	3	0	0	1	0
2	3	0	0	1	0
4	3	0	0	1	0

As we can see in Table 5, value $K=1$ is not appropriate for processing of the sentence ‘Never buy this nice mobile!’, because negation ‘never’ would be in a combination different from the combination comprising the word ‘nice’, to which the negation is related. Setting $K=1$ represents processing of words in isolation from each other. The alternative $K=2$ allows processing of neighbouring words as combinations, but it does not prevent the isolation of negation from relating word either. This sentence can be satisfactorily processed only when the coefficient has value $K \geq 4$.

4.5 Polarity Determination of Word Combinations

Generation of suitable word combinations using the dynamic coefficient K is the key factor of effective opinion classification. These combinations are sets words (their cardinality differs according to changing value of K), to which a polarity degree, representing the polarity of the word combination as a whole, is assigned. This polarity degree is an integer from the set $\{-3, -2, -1, 1, 2, 3\}$. For example, the polarity degree 2 in the second column of the Table 6 can be interpreted as strong positive polarity (SP) or weak positive polarity modified by intensity (WP + I). This intensity is introduced into the given combination by another word, which can precede or follow the word with weak positive polarity. Table 6 illustrates examples of most often used word combinations for K from 2 to 4 together with their interpretation and resulting polarity degree.

Table 6. Polarity degree determination of words combinations with various code lengths (SP+I is Strong Positive + Intensity, SP or WP+I represents Strong Positive or Weak Positive + Intensity and WP is Weak Positive. Similarly, it holds for negative polarity.)

Interpretation	SP + I	SP or WP + I	WP	WN	SN or WN + I	SN + I
K = 2	48	80, 41	10, 32, 23	20, 31, 13	90, 42	49
K = 3	480,408	800, 410, 401	100, 320, 230, 302, 203	200, 310, 130, 301, 103	900, 420, 402	490, 409
K = 4	4800, 4080, 4008	8000, 4100, 4010, 4001	1000, 3200,2300, 3020,2030, 3002,2003	2000, 3100,1300, 3010,1030, 3001,1003	9000, 4200, 4020, 4002	4900, 4090, 4009
polarity	3	2	1	-1	-2	-3

According to the second column of the Table 6, the polarity degree 2 (with its interpretation SP or WP + I) for $K=4$ represents two basic alternatives. The first possible alternative is represented by a strong positive word (8), which is complemented by neutral words (8000). The second possibility is a weak positive word (1) followed (within the same combination) by word increasing polarity intensity (4) and they are complemented by two neutral words in order to form a combination of the given length (4100). These words having non-zero code can be differently ordered within the given word combination (e.g. 4010, 4001).

Table 6 is not presented in its entirety. It only illustrates the most often employed combinations. For example, the second column can be completed with other combinations, for example a weak positive word can be followed by a word increasing polarity intensity (1400, 1040 and 1004).

5 Implementation of the Opinion Classification Method

The presented design of the method of opinion classification has been implemented as well. The implementation within OCS (Opinion Classification System) was used to experiment with the designed method. The OCS is a server application with two interfaces – one interface for “guest” users and another one for “admin” users. Expected competencies of the guest users are: initialization of opinion classification of a selected text and changing the value of the dynamic coefficient, if it is necessary. The admin user has the same competencies as guest but he/she can also create and edit the classification vocabulary. When the OCS system detects a new word within the processed text, it offers to admin the possibility to insert this new word into the classification vocabulary. The admin can decide whether to insert this unknown word (the word has subjectivity) into the vocabulary or not (the word has no subjectivity). This implementation has been realized in the programming language PHP and it is available on the URL <http://mk51.wz.cz/>. More information about this implementation can be found in [7].

The implementation was tested on the set of discussion contributions from the portal <http://www.mobilmania.sk>. This portal focuses on mobile telephones evaluation. Our tests were focused on the discussion thread related to reviews of the mobile telephone LGKU990. The set of contributions used for testing purposes contained 1558 words and 236 lexical units (combinations). The structure of the classification vocabulary was the following: 27 positive words, 27 negative words, 10 negations and 11 words, which increased the intensity of polarity. The evaluation was based on the comparison of results achieved by the OCS system and results obtained from an expert. The expert provided logical analysis of contributions taking into account the structure and meaning of particular sentences. The resulting precision of the implementation OCS according to introduced tests was 78,2%, which is arithmetical average of precision of OCS on positive contributions (86,2%) and on negative contributions (69,2%), what can be seen in Table 7.

Table 7. Results of experiments with the implementation OCS

	OCS result	Expert result	Precision
positive	29	25	0,862
negative	26	18	0,692

We can see in the table, that the OCS implementation classified some neutral or even negative (positive) contribution to the positive (negative) opinion category. There are 4 mistakes in the classification of 29 contributions as positive opinions. For example, the sentence ‘Also my old Sony Ericsson makes *better* photos’ was classified to positive opinion category because of the positive word ‘better’ and lack of ability of OCS to identify hidden irony of this sentence.

The opinion classification is sometimes very complicated not only due to the irony. Complications can arise from indirectly expressed opinion as well. For example, let us consider the sentence ‘I would not buy other brand’. It contains only neutral words and negation without positive or negative word, which this negation is related to. Therefore, the OCS classified this sentence to the neutral opinion class.

6 Conclusions

The automatic opinion classification definitely belongs to up-to-day research agenda. There is a great potential of using the opinion classification within web discussion portals as a service not only for ordinary users (consumers) but for business-entities or organizations (Internet marketing) as well. The application of opinion classification can offer help supporting common users in decision making. Similarly, it can offer some services to business-entities and organizations (e.g. political parties, subjects of civil services, printed and electronic media, marketing agencies, etc.), for example the prediction of the development of society feelings or measuring degree of freedom of media. From this point of view, it is very promising research field.

The achieved precision of our implementation (78,2%) can be perceived as a relatively good result considering the beginning stage of development. During next

research stage, this implementation should be improved in order to perform deeper analysis of the given text and to provide more precise opinion classification. Higher precision can be achieved by means of irony and ambiguity detection. Also, it would be appropriate to test the improved implementation within the more extensive testing environment setting.

Acknowledgements. This work was supported by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic within project 1/0042/10 “Methods for identification, annotation, search, access and composition of services using semantic metadata in support of selected process types”(50%). This work is also the result of the project implementation Development of the Centre of Information and Communication Technologies for Knowledge Systems (project number: 26220120030) supported by the Research & Development Operational Program funded by the ERDF (50%).

References

1. Barla, M., Bieliková, M.: On Deriving Tagsonomies: Keyword Relations Coming from Crowd. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS(LNAI), vol. 5796, pp. 309–320. Springer, Heidelberg (2009)
2. Catena, A., Alexandrov, M., Ponomareva, N.: Opinion Analysis of Publications on Economics with a Limited Vocabulary of Sentiments. *International Journal on Social Media. MMM: Monitoring, Measurement, and Mining* 1(1), 20–31 (2010)
3. Ding, X., Liu, B., Yu, A.P.: Holistic Lexicon-Based Approach to Opinion Mining. In: Proc. of the Int. Conf. on Web Search and Web Data Mining, WSDM 2008, New York, NY, USA, pp. 231–240 (2008)
4. Heer, J., Boyd, D.: Vizster: Visualizing Online Social Networks. In: Proceedings of the IEEE Symposium on Information Visualization INFOVIS 2005, Washington, USA, pp. 5–13 (2005)
5. Kaurova, O., Alexandrov, M., Ponomareva, N.: The Study of Sentiment Word Granularity for Opinion Analysis (a Comparison with Maite Taboada Works). *International Journal on Social Media. MMM: Monitoring, Measurement, and Mining* 1(1), 45–57 (2010)
6. Lukáč, G., Butka, P., Mach, M.: Semantically-enhanced extension of the discussion analysis algorithm in SAKE. In: SAMI 2008, 6th International Symposium on Applied Machine Intelligence and Informatics, Herľany, Slovakia, pp. 241–246 (January 2008)
7. Machová, K., Krajč, M.: Opinion Classification in Threaded Discussions on the Web. In: Proc. of the 10th Annual International Conference Znalosti 2011, Stará Lesná, pp. 136–147. FEI Technická univerzita Ostrava, Czech Republic (2011)
8. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. *Foundation and Trends in Information Retrieval* 2(1-2), 1–135 (2008)

Modelling Trust for Communicating Agents: Agent-Based and Population-Based Perspectives

S. Waqar Jaffry and Jan Treur

VU University Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
{s.w.g.jaffry, j.treur}@vu.nl
<http://www.few.vu.nl/~{swjaffry, treur}>

Abstract. This paper presents an exploration of the differences between agent-based and population-based models for trust dynamics. This exploration is based on both a large variety of simulation experiments and a mathematical analysis of the equilibria of the two types of models. The outcomes show that the differences between the models are not very substantial, and become less for larger numbers of agents.

Keywords: agent-based, population-based, trust.

1 Introduction

When a population of agents is considered, the dynamics of trust in a certain trustee can be modelled from two perspectives: from the agent-based perspective and from the population-based perspective. From the agent-based perspective each agent has its own characteristics and maintains its own trust level over time. From the population-based perspective one trust level for the whole is maintained over time, depending on characteristics of the population. For both cases dynamical models can be used to determine the trust levels over time. For the agent-based perspective, each agent has its own dynamical model (for example, expressed as a system of N differential equations, with N the number of agents), whereas for the population-level one model (for example, expressed as one differential equation) can be used. From the agent-based model, by aggregation a collective trust level for the population as a whole can be determined, for example, by taking the average over all agents.

Usually agent-based simulation is computationally much more expensive than population-based simulation. However, this still may be worth the effort when it is assumed that the outcomes substantially differ from the outcomes of a population-based simulation. In this paper this assumption is explored in a detailed manner for a population of agents that not only receive direct experiences for a trustee but also get communicated information from other agents about their trust in this trustee. On the one hand the analysis makes use of a variety of simulation experiments for different population sizes and different distributions of characteristics. On the other hand a mathematical analysis of equilibria of both types of models is used to find out differences between the two types of models. Roughly spoken, the outcome of both

type of investigations are that in general the differences are not substantial, and that they are smaller the larger the number of agents is.

In Section 2 the two types of models used are introduced. In Section 3 the simulation experiments are described. Section 4 presents the mathematical analysis of equilibria. Section 5 concludes the paper.

2 Modelling Trust Dynamics from Two Perspectives

In this section trust models for both perspectives are introduced. The basic underlying trust dynamics model adopted in this paper depends on receiving experiences $E(t)$ over time as follows:

$$T(t + \Delta t) = T(t) + \gamma * (E(t) - T(t)) * \Delta t$$

Here $T(t)$ and $E(t)$ are the trust level for a trustee and the experience level given by the trustee at time point t . Furthermore, γ is a personal characteristic for flexibility: the rate of change of trust upon receiving an experience $E(t)$. The values of $T(t)$, $E(t)$ and γ are in the interval $[0, 1]$. In differential form this model can be expressed by

$$\frac{dT}{dt} = \gamma * (E - T)$$

This basic model is based on the experienced-based trust model described in [5], and applied in [6, 7, 8]. In the case of communicating agents, experiences are taken to be of two forms: direct experiences acquired, for example, by observation, and indirect experiences, obtained from communication. Incorporating this, the basic model can be applied to each single agent within the population (agent-based perspective), or to the population as a whole (population-based perspective), as discussed below.

2.1 An Agent-Based Trust Model Incorporating Communication

In the agent-based trust model described here, each of the agents updates its trust on a given trustee based on receiving an experience for this trustee which combines a direct experience and an opinion received by the peers about the trustee (indirect experience). Direct and indirect experiences at each time point are aggregated using the agents' personality characteristic called social influence denoted by α_A as follows:

$$E_A(t) = \alpha_A * E_A^i(t) + (1 - \alpha_A) * E_A^d(t)$$

Here $E_A(t)$, $E_A^d(t)$ and $E_A^i(t)$ are the aggregated experience, the direct experience received from the trustee and the indirect experience received by the agent A as the opinions of its peers at about trustee at time t respectively.

The indirect experience $E_A^i(t)$ received by the agent A about a trustee at time point t is taken the average of the opinions given by all the peers at time point t :

$$E_A^i(t) = \sum_{B \neq A} O_B(t) / (N - 1)$$

Here $O_B(t)$ is the opinion received by the agent A from an agent B about the trustee at time point t and N is the total number of agents in the population. The opinion given by the agent B to the agent A at time t is taken as the value of the trust of B on trustee

at time t , so $O_B(t) = T_B(t)$. The aggregated experience received by agent A at time point t is used to update current trust level of the agent A at trustee using trust model presented in the previous section as follows

$$T_A(t + \Delta t) = T_A(t) + \gamma_A * (E_A(t) - T_A(t)) * \Delta t$$

Here the basic trust model is indexed for each agent A in the group. Note that each agent can have its personal flexibility characteristic γ_A . It is assumed that these values have some distribution over the population. Based on this agent-based model a collective trust value $T_C(t)$ for the population as a whole can be obtained by aggregation of the trust values over all agents (taking the average):

$$T_C(t) = \frac{1}{N} \sum_A T_A(t).$$

2.2 A Population-Based Trust Model Incorporating Communication

To apply the basic trust model to obtain a population-based model of trust, its ingredients have to be considered for the population P as a whole, for example, the (direct and indirect) experience given by the trustee to a population P , and the characteristics γ_P of the population [2, 3]; this is done as follows

$$T_P(t + \Delta t) = T_P(t) + \gamma_P * (E_P(t) - T_P(t)) * \Delta t$$

Here $T_P(t)$ is the trust of population P on a given trustee at time point t , and the population-level flexibility characteristic γ_P is taken as an aggregate value for the individual flexibility characteristics γ_A for all agents A in P (e.g., the average of the γ_A for $A \in P$). This can be interpreted as if the population as a whole is represented as one agent who receives experiences from the trustee and updates its trust on the trustee using the basic model. The experience at population level $E_P(t)$ at time point t for the population P is defined as the combination of the direct and the indirect experience at population level as follows,

$$E_P(t) = \alpha_P * E_P^i(t) + (1 - \alpha_P) * E_P^d(t)$$

In the above equation $E_P^i(t)$ and $E_P^d(t)$ are the indirect and direct experience at the population level. Moreover, α_P is the population-level social influence characteristic. Here also α_P is taken as an aggregate value for the individual social influence characteristics α_A for all agents present in P (e.g., the average of the α_A for $A \in P$). At the population level the indirect experience $E_P^i(t)$ obtained from communication by the other agents of their trust is taken as the population level trust value at time point t as follows:

$$E_P^i(t) = T_P(t)$$

2.3 Complexity Estimation

The complexity of the agent-based trust model differs from that of the population-based models in the sense that for the agent-based trust model the complexity depends on the number of agents while this is not the case for the population-based model. This can be estimated as follows. For τ the total number of time steps, and N the number of agents in the population, the time complexities of the agent-based and

population-based models are $O(N^2\tau)$ and $O(\tau)$ respectively. This indicates that for higher numbers of agents in a population the agent-based model is computationally much more expensive.

3 Simulating and Comparing the Two Trust Models

A number of simulation experiments have been conducted to compare the agent-based and population-based trust models as described in the previous sections. This section presents the experimental setup and results from these experiments.

3.1 The Experimental Setup

For the simulation experiments a setup was used as shown in Fig. 1. Here a trustee S is assumed to give similar direct experiences $E^d(t)$ to both models at each time point t . In the population-based trust model this direct experience $E^d(t)$ is used together with the indirect experience $E_p^i(t)$ to update the population-level trust of S according to the equations presented in Section 2.2. In the agent-based trust model this experience is received by every agent in the system and each agent updates its trust on the trustee using direct experience $E^d(t)$ and indirect experience $E_A^i(t)$ received as opinion of the other agents, as shown in Section 2.1. By aggregation the individual trust levels can be used to obtain a collective trust of the trustee.

In Fig. 1 P carries the population-based trust model while the agents $A_1, A_2, A_3 \dots A_n$ carry the agent-based trust model as described in the previous sections. Every agent in the system is assigned an initial trust value $T_A(0)$, a value for the agent's flexibility γ_A , and for the social influence parameter α_A at the start of the simulation experiment. The value $T_p(0)$ for the initial population-level trust, the population-level flexibility parameter γ_p and the social influence α_A parameter for the population-based trust model are taken as the average of the corresponding attributes of all the agents in the community:

$$T_p(0) = \sum_A T_A(0) / N \quad \gamma_p = \sum_A \gamma_A / N \quad \alpha_p = \sum_A \alpha_A / N$$

Here N is the total number of agents in the community. The collective trust of the agent-based trust model at any time point t is represented as the average of the trust values of all the agents in the community:

$$T_C(t) = \sum_A T_A(t) / N$$

As a measure of dissimilarity for the comparison of the models their root mean square error is measured between the collective agent-level trust and population-level trust at each time point t as follows

$$\varepsilon = \sqrt{\sum_{t=1}^{t=M} (T_p(t) - T_C(t))^2 / M}$$

In the above equation $T_p(t)$ and $T_C(t)$ are the population-level trust and the (aggregated) collective agent-level trust of the trustee calculated by the population-based and agent-based model at time point t respectively and M is the total time steps in the simulation.

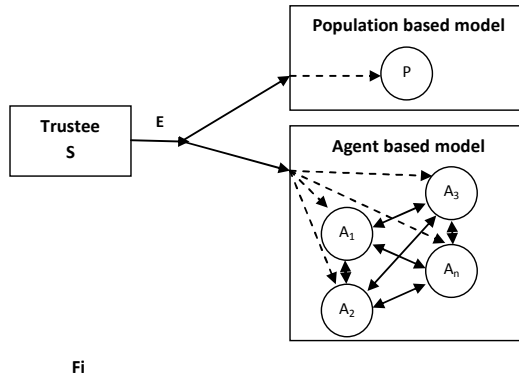


Fig. 1. Agent-based and population-based trust model

To produce realistic simulations, the values for $T_A(0)$, γ_A and α_A of all the agents in the agent-based trust model were taken from a uniform normal distribution with mean value 0.50 and standard deviation varying from 0.00 to 0.24. In these experiments all agent-based models were simulated exhaustively to see their average behavior against the population-based model. Here exhaustive simulation means that all possible combinations of standard deviations for $T_A(0)$, γ_A and α_A from the interval 0.00-0.24 were used in the simulations and their respective errors were measured against respective population level model. An average error ϵ_{avg} of the models was calculated, which is the average of all root mean squared errors calculated with all combinations of $T_A(0)$, γ_A and α_A as follows.

$$\epsilon_{avg} = \frac{\sum_{stDev_{T_A(0)}=0.00}^{stDev_{T_A(0)}=0.24} \left(\sum_{stDev_{\gamma_A}=0.00}^{stDev_{\gamma_A}=0.24} \left(\sum_{stDev_{\alpha_A}=0.00}^{stDev_{\alpha_A}=0.24} \left(\epsilon(stDev_{T_A(0)}, stDev_{\gamma_A}, stDev_{\alpha_A}) \right) \right) \right)}{15625}$$

In the above equation $stDev_{T_A(0)}$, $stDev_{\gamma_A}$ and $stDev_{\alpha_A}$ are the standard deviation values used to generate the agents’ initial trust values, the agents’ trust flexibility parameter, and agents’ social influence parameter from a uniform normal distribution around the mean value of 0.50. Here $\epsilon(stDev_{T_A(0)}, stDev_{\gamma_A}, stDev_{\alpha_A})$ is the error calculated for an experimental setup where $T_A(0)$, γ_A , and α_A were taken using $stDev_{T_A(0)}$, $stDev_{\gamma_A}$ and $stDev_{\alpha_A}$ as standard deviation for a random number generator. Here it can be noted that to obtain the average, this summation is divided by 15625 which are the number of comparison models generated by all variations in $stDev_{T_A(0)}$, $stDev_{\gamma_A}$, and $stDev_{\alpha_A}$, e.g. $25*25*25$.

In order to simulate realistic behavior of the trustee’s experience E to the agents, E was also taken from a uniform normal distribution with mean value of 0.50 and experience’s standard deviation $stDev_E$ from the interval 0.00 – 0.24. These experience values were also taken exhaustively over $stDev_{T_A(0)}$, $stDev_{\gamma_A}$, and $stDev_{\alpha_A}$. The algorithm for the simulation experiments is presented below; it compares the population-based trust model with the agent-based trust model exhaustively with all possible standard deviations of $stDev_E$, $stDev_{\gamma_A}$, $stDev_{T_A(0)}$ and $stDev_{\alpha_A}$ varying in the interval 0.00-0.24 described as follows.

Algorithm S: Agent and population base model comparison

```

00: Agent [ $A_1, A_2, \dots, A_n$ ] of ABM, Agent  $P$  of PBM, Trustee  $S$ ;
01: for all  $stdDev_E$  from 0.00 to 0.24
02:   for all  $stdDev_{\gamma_A}$  from 0.00 to 0.24
03:     for all  $stdDev_{T_A(0)}$  from 0.00 to 0.24
04:       for all  $stdDev_{\alpha_A}$  from 0.00 to 0.24
05:         for all Agents  $A$  in ABM
06:           initialize  $T_A(0)$  of  $A$  from  $stdDev_{T_A(0)}$ 
07:           initialize  $\gamma_A$  of  $A$  from  $stdDev_{\gamma_A}$ 
08:           initialize  $\alpha_A$  of  $A$  from  $stdDev_{\alpha_A}$ 
09:         end for [all agents  $A$ ]
10:       initialize  $T_P(0)$ ,  $\gamma_P$  and  $\alpha_P$  of  $P$  with average of  $T_A(0)$ ,  $\gamma_A$  and  $\alpha_A$ 
11:       for all time points  $t$ 
12:         trustee  $S$  gives experience  $E(t)$  from  $stdDev_E$ 
13:         agent  $P$  receives  $E_P^d(t)$  and calculates  $E_P^i(t)$  where  $E_P^d(t) = E(t)$ 
14:         agent  $P$  updates trust  $T_P(t)$  of  $S$ 
15:         for all agents  $A$  in ABM
16:            $A$  receives experience  $E_A^d(t)$  where  $E_A^d(t) = E(t)$ 
17:           for all agents  $B$  in ABM where  $A \neq B$ 
18:              $A$  gets opinion  $O_{AB}(t)$  from  $B$  and aggregate in  $E_A^i(t)$ 
19:           end for [all agents  $B$ ]
20:            $A$  updates trust  $T_A(t)$  on  $S$ 
21:           update  $T_C(t)$  of  $S$  using trust  $T_A(t)$  of  $A$ 
22:         end for [all agents  $A$ ]
23:         calculate error  $\varepsilon$  of models using  $T_P(t)$  and  $T_C(t)$ 
24:       end for [all time points  $t$ ]
25:     end for [all agents  $stdDev_{\alpha_A}$ ]
26:   end for [all  $stdDev_{T_A(0)}$ ]
27:   calculate average models error  $\varepsilon_{avg}$  for all models( $stdDev_{\gamma_A}$ ,  $stdDev_{T_A(0)}$ ,  $stdDev_{T_A(0)}$ )
28: end for [all  $stdDev_{\gamma_A}$ ]
29: calculate average experience level error  $\varepsilon_E$  for all experience sequences using  $\varepsilon_{avg}$ 
30: end for [all  $stdDev_E$ ]

```

3.2 Experimental Configurations

In Table 1 the experimental configurations used for the different simulations are summarized. All simulations were run for 500 time steps, and were performed for different values for the agents in the agent-based model to cover different types of populations. The parameter SS for the sample of simulation experiments is taken 25: each experiment is run 25 times after which an average is taken. This is meant to undo the randomization effects and to get the general average characteristics of the models. To obtain a wide variety of possible dynamics of the agent-based trust model the agents' initial trust, the agents' flexibility, agents' social influence and the experience with the trustee were taken exhaustively from a uniform normal distribution with various standard deviations.

Given the above experimental configurations, time complexity for the simulation experiments for the algorithm S is $O(stdDev_E \cdot stdDev_{\gamma} \cdot stdDev_{T_A(0)} \cdot stdDev_{\alpha} \cdot TT \cdot N \cdot SS)$. For $stdDev_E$, $stdDev_{\gamma}$, $stdDev_{T_A(0)}$ and $stdDev_{\alpha}$ ranging from 0.00 to 0.24, 500 time steps for simulation, 50 agents and 25 samples of simulation the approximate number for the instruction count becomes 1.22×10^{13} .

Table 1. Experimental configurations

Name	Symbol	Value
Total time steps	TT	500
Number of agents	N	10, 20, 30, 40, 50
Samples of simulation experiments	SS	25
Standard deviation and mean for direct experience	$stdDev_E, mean_E$	0.00-0.24, 0.50
Standard deviation and mean for rate of change	$stdDev_\gamma, mean_\gamma$	0.00-0.24, 0.50
Standard deviation and mean for initial trust	$stdDev_{T(0)}, mean_{T(0)}$	0.00-0.24, 0.50
Standard deviation and mean for social influence	$stdDev_\alpha, mean_\alpha$	0.00-0.24, 0.50

3.3 Simulation Results

The algorithm S specified in Section 3.1 was implemented in the programming language C++ to conduct the simulations experiments using the configuration as described in Table 1, and to compare the agent-based and population-based trust models. In this section some of the simulation results are discussed.

Experiment 1: Variation in the experience value from the trustee

In this experiment an exhaustive simulation was performed where the trustee gives experience values from a uniform normal distribution around the mean value 0.50 with standard deviation $stdDev_E$ from the interval 0.00 to 0.24. For each value of $stdDev_E$ the agents' initial trust, flexibility and social influence parameters were taken from a uniform normal distribution with mean value 0.50 and standard deviation varying from 0.00 to 0.24 (see algorithm S). To see the effect of the population size on this experiment, the experiment was executed for different numbers of agents varying from 10 to 50. Some of the results are shown in Fig. 2.

In Fig. 2a) the horizontal axis represents the standard deviation in the experience values E given by the trustee, varying from 0.00 to 0.24 and the vertical axis shows the average experience level error ε_E of all models with standard deviations of the agent attributes $T_A(0)$, γ_A and α_A in the agent-based model, varying from 0.00 to 0.24. Here it can be seen that upon an increase in standard deviation of experience value given by the trustee, the average error between the agent-based and population-based model increases for all population sizes (from about 0.001 to about 0.004). This error values is lower for higher numbers of agents which shows that the population-based model is a much better approximation of the agent-based based model for higher number of agents. In Fig. 2b) the horizontal axis shows the number of agents in the agent-based model while the vertical axis represents the average of the experience level error ε_E for all models, where the trustee gives experience values with standard deviation $stdDev_E$ (varying from 0.00 to 0.24), and the agents in the agent-based model have attributes $T_A(0)$, γ_A and α_A with standard deviations $stdDev_{\gamma_A}$, $stdDev_{T_A(0)}$, and $stdDev_\alpha$ (varying from 0.00 to 0.24). Here it can also be observed that the population-based trust model provides a (slightly) more accurate approximation of the agent-based model, when having larger numbers of agents (from about 0.0026 to about 0.0024).

In all these experiments the maximum root mean squared error between agent-based and population-based trust model does not exceed 0.027267, which means that this population-based trust model is a quite accurate approximation of the agent-based model.

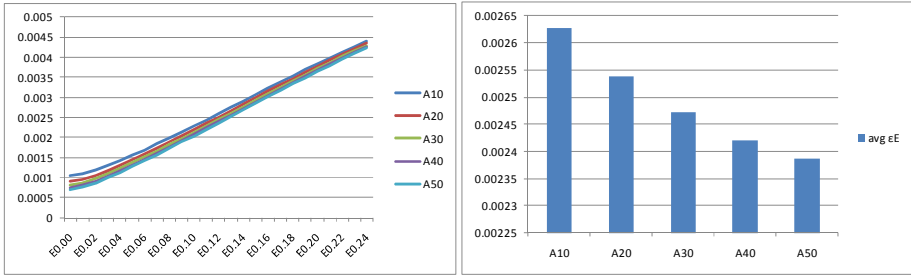


Fig. 2. a) Difference between the two models upon variation in experience values b) Average difference (error) between the agent-based and population-based trust models for all possible standard deviations of $stdDev_{\gamma}$, $stdDev_{T_A(0)}$, $stdDev_{\alpha}$ and $stdDev_E$

Experiment 2: Exhaustive mirroring of agent-based into population-based model

In the previous experiment the attribute values of the population-level model were simply taken as an average of the attribute values of all agents in the agent-level model. However, it cannot be claimed at forehand that this mechanism of abstracting the agent-level model is the most accurate aggregation technique. In order to see whether there is any other instance of the population-level model that can approximate the agent-level models better than the one based on aggregating by averaging, one has to exhaustively simulate all instances of the population-based model against all instances of the agent-based model. In this experiment such an exhaustive simulation was performed, applying a method named as *exhaustive mirroring of models*, adopted from [4]. In this method of mirroring of models the target model is exhaustively (for different parameter settings) simulated to realize a specific trace of the source model for a given set of parameters of source model. The instance of the target model for specific values of the parameters that generate a minimal error is considered as the best realization of the target model to approximate the source model. As stated in [4] this process gives some measure of similarity of the target model against the source model. However, this method of exhaustive mirroring is computationally very expensive. So, for practical reasons in this experiment the population-based model (target) is exhaustively simulated with only one of the three population level parameters, namely the flexibility γ_p of the population-level trust. The other two parameters the population-level (initial trust $T_p(0)$ and social influence α_p) were taken as the average of their counterparts in the agent-level model. Some of the results of this experiment are shown in Fig. 3; In Fig. 3a) the horizontal axis represents the exhaustive values for the population-level flexibility parameter γ_p and the vertical axis shows the average experience level error ϵ_E of all agent-based models with standard deviations of the attributes $T_A(0)$, γ_A , α_A and trustee experience $E^d(t)$ varying from 0.00 to 0.24 with mean value 0.5. Here it can be seen that for lower values of γ_p the average error is much higher and it starts to reduce when γ_p approaches to 0.5 and values of γ_p above 0.5 this error starts to increase. Hence 0.50 is the most accurate representation of γ_p for all agent base models. Further in Fig. 3b) same graph is shown in a zoomed-in fashion to show the effect of population size on error value. Here it is seen that larger populations showed lower error than smaller populations.

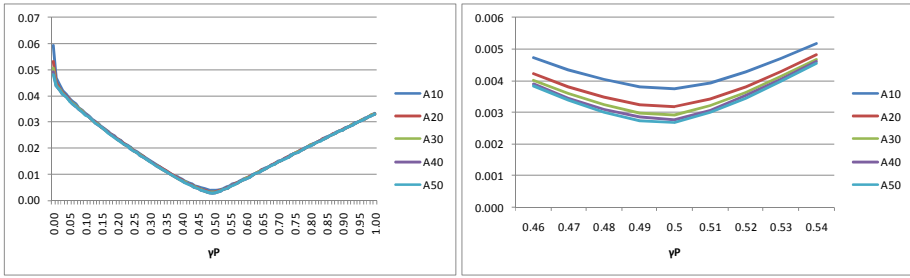


Fig. 3. a) Difference between agent-based and population-based trust models upon change in population level flexibility parameter γ_p , **b)** Zoomed-in version of Fig. 3a)

Experiment 3: Comparison for larger populations with numbers up to 500 agents

Based on observation from the experiments described above some support was obtained that the value 0.5 for the population-level flexibility parameter γ_p is the most accurate representation of the agent-based model. To get a better impression for the limit value of the error for larger populations, in the next experiment the agent-based model were simulated for larger populations up to 500 agents in size and compared to the population-based model with flexibility parameter $\gamma_p = 0.5$. In this experiment the population was varied from 200 to 500 agents with an increment of 10 agents per population size. Experimental configurations in this experiment were taken from Table 1. Results are shown in Fig. 4; In Fig. 4a) the horizontal axis represents the different population sizes varying from 200 to 500 agents and the vertical axis shows the average difference between agent and population level models. Here it can be seen that on an increase in number of agents in population base model difference between models decreases from about 0.00229 (for 200 agents) to about 0.00218 (for 500 agents). It has been analysed in how far the approximation of the limit value for the error for larger populations is exponential and how the limit value can be estimated from the obtained trend. To this end Fig. 4b) depicts for a certain value of e (an assumed limit value) the graph of the logarithm of the distance of the error to e , expressed as $\ln(\text{error} - e)$. This graph (in blue) is compared to a straight line (in red). It turns out that in 6 decimals the straight line is approximated best for limit value $e = 0.002145$, and the approximation of this limit value for e goes exponentially according to an average (geometric mean) factor 0.947149 per increase of 10 agents.

In summary, given that the error found for $N = 200$ is 0.002288, based on this extrapolation method the difference between the agent-based and population-based model for larger population sizes $N \geq 200$ can be estimated as

$$\begin{aligned} est_error(N) &= 0.002145 + (0.002288 - 0.002145) * 0.947149^{N-200} \\ &= 0.002145 + 0.000143 * 0.947149^{N-200} \end{aligned}$$

This estimation predicts that always an error of at least 0.002145 is to be expected; this actually is quite low, but it will not become still lower in the limit for very large N . It turns out that the difference between actual error and estimated error using the above formula for all N between 200 and 500 is less than $2 \cdot 10^{-6}$, with an average of $7 \cdot 10^{-7}$. Note that by having this estimation of the error, it can also be used to correct the population-based model for it, thus in a cheap manner approximating the agent-based model by an accuracy around 10^{-6} .

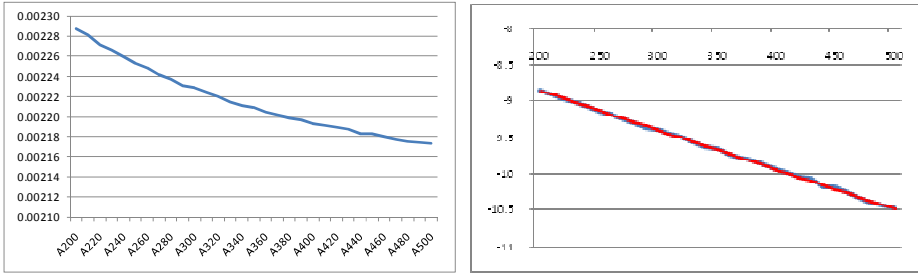


Fig. 4. a) Difference between agent-based and population-based trust models upon change in population size on level flexibility parameter $\gamma_p = 0.5$
 b) Graph of $\ln(\text{error} - e)$ compared to a straight line for $e = 0.002145$

Computational time complexity of experiments

The computation complexities of the experiments described above are primarily based on the complexity estimations presented in Section 3.2. It was clear that the nature of these experiments was exhaustive, hence to conduct them on a desktop PC would require a large amount of time. So these experiments were conducted on the Distributed ASCII Supercomputer version 4, DAS-4 [1]. The computation complexity of these experiments is shown in Fig. 5. In this figure the horizontal axis represents the different experiments described in the previous sections and the vertical axis shows the number of days required to complete these experiments on a PC and on DAS-4. Here it should be noted that for experiment 1 and 3, 25 nodes of DAS-4 are used while for experiment 2, 100 nodes of DAS-4 were utilized. As it can be seen from Figure 5 all three experiments were expected to take approximately 55 days on a single machine while usage of DAS-4 has reduced this time to approximately 1.25 days.

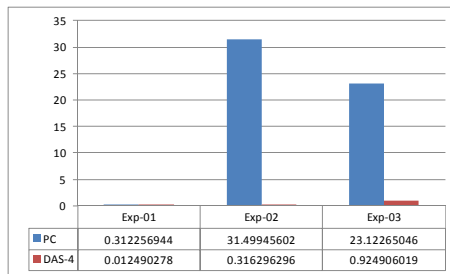


Fig. 5. Computation time required for the simulation experiments on PC and DAS-4

4 Mathematical Analysis of Equilibria of the Two Models

The agent-based and population-based models can also be analysed mathematically by determining equilibria. These are values for the variables upon which no change occurs anymore. For equilibria also the externally given experience values have to be constant; instead of these values for E also the expectation value for them can be taken. For the population-level model, assuming flexibility $\gamma_p > 0$ an equilibrium has to satisfy $T_P(t) = E_P(t)$ with

$$E_P(t) = \alpha_P T_P(t) + (1 - \alpha_P) E_P^d(t)$$

Leaving out t , and taking $E = E_P^d$, this provides the following equation in T_P

$$T_P = \alpha_P T_P + (1 - \alpha_P) E$$

Thus (assuming $\alpha_P \neq 1$) an equilibrium $T_P = E$ is obtained. In a similar manner for the agent-based model equilibria can be determined. Again, assuming flexibility $\gamma_A > 0$ an equilibrium has to satisfy $T_A(t) = E_A(t)$ for each agent A , this time with

$$E_A(t) = \alpha_A E_A^i(t) + (1 - \alpha_A) E_A^d(t)$$

where $E_A^i(t) = \sum_{B \neq A} T_B(t)/(N-1)$. This provides N equations

$$T_A(t) = \alpha_A \sum_{B \neq A} T_B(t)/(N-1) + (1 - \alpha_A) E$$

Aggregating these equations, and leaving out t , the relation to collective trust is found:

$$\begin{aligned} \sum_A T_A/N &= \sum_A [\alpha_A \sum_{B \neq A} T_B/(N-1) + (1 - \alpha_A) E] / N \\ T_C &= \sum_A \alpha_A \sum_{B \neq A} T_B/(N-1)N + \sum_A (1 - \alpha_A) E / N \\ &= \sum_A \alpha_A [\sum_B T_B - T_A]/(N-1)N + (1 - \sum_A \alpha_A/N) E \\ &= [\sum_A \alpha_A \sum_B T_B - \sum_A \alpha_A T_A]/(N-1)N + (1 - \sum_A \alpha_A/N) E \\ &= [\sum_A \alpha_A T_C / (N-1) - \sum_A \alpha_A T_A / (N-1)N] + (1 - \sum_A \alpha_A/N) E \\ &= [(\sum_A \alpha_A / N) T_C N / (N-1) - \sum_A \alpha_A T_A / (N-1)N] + (1 - \sum_A \alpha_A/N) E \\ &= [(\sum_A \alpha_A / N) T_C + (\sum_A \alpha_A / N) T_C / (N-1) - \sum_A \alpha_A T_A / (N-1)N] + (1 - \sum_A \alpha_A/N) E \\ &= (\sum_A \alpha_A / N) T_C + (1 - \sum_A \alpha_A/N) E + [(\sum_A \alpha_A / N) T_C / (N-1) - \sum_A \alpha_A T_A / (N-1)N] \\ &= (\sum_A \alpha_A / N) T_C + (1 - \sum_A \alpha_A/N) E + \sum_A \alpha_A [T_C - T_A] / (N-1)N \end{aligned}$$

So, taking $\alpha_C = \sum_A \alpha_A / N$ the following equilibrium equation is obtained:

$$\begin{aligned} (1 - \alpha_C) T_C &= (1 - \alpha_C) E + \sum_A \alpha_A [T_C - T_A] / (N-1)N \\ T_C &= E + \sum_A \alpha_A [T_C - T_A] / (N-1)N(1 - \alpha_C) \end{aligned}$$

Therefore in general the difference between the equilibrium values for T_C (aggregated agent-based model) and T_P (population-based model) can be estimated as

$$T_C - T_P = T_C - E = \sum_A \alpha_A [T_C - T_A] / (N-1)N(1 - \alpha_C)$$

As T_C and T_A are both between 0 and 1, the absolute value of the expression in $T_C - T_A$ can be bounded as follows

$$|\sum_A \alpha_A [T_C - T_A] / (N-1)N(1 - \alpha_C)| \leq \sum_A \alpha_A / (N-1)N(1 - \alpha_C) = \alpha_C / (N-1)(1 - \alpha_C)$$

Therefore the following bound for the difference in equilibrium values is found:

$$|T_C - T_P| \leq \alpha_C / (N-1)(1 - \alpha_C)$$

This goes to 0 for large N , which would provide the value $T_C = E = T_P$. For $\alpha_C = 0.5$, and $N = 200$, this bound is about 0.005, for $N = 500$, it is about 0.002. These deviations are in the same order of magnitude as the ones found in the simulations. Note that the expression in $T_C - T_A$ also depends on the variation in the population. When all agents have equal characteristics $\alpha_A = \alpha$ it is 0, so that $T_C = E = T_P$.

$$\begin{aligned} T_C - T_P &= \alpha \Sigma_A [T_C - T_A] / (N-1)N(1-\alpha_C) = \alpha [\Sigma_A T_C / N - \Sigma_A T_A / N] / (N-1) (1-\alpha_C) \\ &= \alpha [T_C - T_A] / (N-1)(1-\alpha_C) = 0 \end{aligned}$$

So also in the case of equal parameter values for α_A it holds $T_C = E = T_P$; note that this is independent of the variation for the other parameters.

5 Conclusion

This paper addressed an exploration of the differences between agent-based and population-based models for trust dynamics, based on both a large variety of simulation experiments and a mathematical analysis of the equilibria of the two types of models.

By both types of exploration it was shown that the differences between the two types of model are quite small, in general below 1%, and become less for larger numbers of agents. An implication of this is that when for a certain application such an accuracy is acceptable, instead of the computationally more expensive agent-based modelling approach (complexity $O(N^2\tau)$ with N the number of agents and τ the number of time steps), as an approximation also the population-based approach can be used (complexity $O(\tau)$).

The experiments to find these results were conducted on the Distributed ASCI Supercomputer version 4, DAS-4 [1], thereby using 25, resp. 100 processors. The experiments were expected to take approximately 55 days on a single PC; the use of DAS-4 has reduced this time to approximately 1.25 days.

References

1. Bal, H., Bhoedjang, R., Hofman, R., Jacobs, C., Kielmann, T., Maassen, J., et al.: The distributed ASCI Supercomputer project. *SIGOPS Oper. Syst. Rev.* 34, 76–96 (2000)
2. Hoogendoorn, M., Jaffry, S.W.: The Influence of Personalities Upon the Dynamics of Trust and Reputation. In: *Proc. Computational Science and Engineering, CSE 2009*, pp. 263–270 (2009)
3. Huff, L., Kelley, L.: Levels of Organizational Trust in Individualist versus Collectivist Societies: A Seven Nation Study. *Organizational Science* 14, 81–90 (2003)
4. Jaffry, S.W., Treur, J.: Comparing a Cognitive and a Neural Model for Relative Trust Dynamics. In: Leung, C., Lee, M., Chan, J. (eds.) *ICONIP 2009*. LNCS, vol. 5863, pp. 72–83. Springer, Heidelberg (2009)
5. Jonker, C.M., Treur, J.: A Temporal-Interactivist Perspective on the Dynamics of Mental States. *Cognitive Systems Research Journal* 4, 137–155 (2003)
6. Singh, S.I., Sinha, S.K.: A New Trust Model Based on Time Series Prediction and Markov Model. In: Das, V.V., Vijaykumar, R. (eds.) *ICT 2010*. CCIS, vol. 101, pp. 148–156. Springer, Heidelberg (2010)
7. Skopik, F., Schall, D., Dustdar, S.: Modeling and mining of dynamic trust in complex service-oriented systems. *Information Systems* 35, 735–757 (2010)
8. Walter, F.E., Battiston, S., Schweitzer, F.: Personalised and Dynamic Trust in Social Networks. In: Bergman, L., Tuzhilin, A., Burke, R., Felfernig, A., Schmidt-Thieme, L. (eds.) *Proceedings of the Third ACM conference on Recommender systems, RecSys 2009*, pp. 197–204. ACM Press, New York (2009)

Multidimensional Social Network: Model and Analysis

Przemysław Kazienko¹, Katarzyna Musiał², Elżbieta Kukła¹, Tomasz Kajdanowicz¹,
and Piotr Bródka¹

¹ Wrocław University of Technology, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
{kazienko, elzbieta.kukla, tomasz.kajdanowicz,
piotr.brodka}@pwr.wroc.pl

² Smart Technology Research Centre, Bournemouth University, Fern Barrow, Poole, Dorset
BH12 5BB, UK
kmusial@bournemouth.ac.uk

Abstract. A social network is an abstract concept consisting of set of people and relationships linking pairs of humans. A new multidimensional model, which covers three main dimensions: relation layer, time window and group, is proposed in the paper. These dimensions have a common set of nodes, typically, corresponding to human beings. Relation layers, in turn, reflect various relationship types extracted from different user activities gathered in computer systems. The time dimension corresponds to temporal variability of the social network. Social groups are extracted by means of clustering methods and group people who are close to each other. An atomic component of the multidimensional social network is a view – small social sub-network, which is in the intersection of all dimensions. A view describes the state of one social group, linked by one type of relationship (one layer), and derived from one time period. The multidimensional model of a social network is similar to a general concept of data warehouse, in which a fact corresponds to a view. Aggregation possibilities and usage of the model is also discussed in the paper.

Keywords: social network, multidimensional social network, multi-layered social network, network model.

1 Introduction

Every social networked system is a source of different information about people and their activities. Lately we have experienced rapid growth of social structures supported by communication technologies and the variety of Internet- and Web-based services. For the first time in the human history we have possibility to process data (gathered in our computer systems) about interactions and activities of millions of individuals. Communication technologies allow us to form large networks which in turn shape and catalyse our activities. Due to scale, complexity and dynamics, these networks are extremely difficult to analyse in terms of traditional social network analysis methods being at our disposal. Moreover, there is hardly any research reported with respect to multidimensional networked models.

Nodes in such complex social networks are digital representations of people who use email services, telecommunication systems, multimedia sharing systems, access blogosphere etc. Based on interactions between users their mutual relationships are extracted. Due to diversity of communication channels the analyzed networks are multidimensional, i.e. these are networks that consist of more than one type of relationship. Different relations can emerge from different communication channels, i.e. based on each communication channel separate relation that can be also called a layer of a network is created. Moreover, there is one more dimension that needs to be considered – time. The behaviour of all nodes in social network is time-dependent, i.e. time factor cannot be neglected during analysis.

Although different models of social networks have been developed and many ways of their representation, such as matrices, graphs or algebraic description, exist, there is hardly any research reported with respect to the modelling of multidimensional social networks. This article focuses on building a model for multidimensional social network that will also be able to present the dynamics of this structure.

2 Related Work

The research in the field of social networks has its origins in 1950s and since then is continuously developed by many scientists.

As it was mentioned before, a social network can be defined as the finite set of actors (network nodes) and relationships (network edges) that link these actors. Although this concept appears to be quite obvious, different researchers describe it in a slightly different way and from different perspectives [11], [18], [20], [21]. These various approaches are the result of the fact that the social network concept has been simultaneously developed by scientists from many different domains.

Based on data gathered in computer systems, a new type of social networks can be extracted and analysed. These networks are named online social networks [5], [8], web-based social networks [10], computer-supported social networks [22] or virtual communities. They can be created from different data sources, e.g.: bibliographic data [9], blogs [1], photos sharing systems like Flickr [14], e-mail systems [16], telecommunication data [2], [17], social services like Twitter [12] or Facebook [6], video sharing systems like YouTube [4], and many more.

During analysis of social networks, researchers usually take into account only one type of connections between users while, in most real cases, there are many different relationships. Only few scientists have focused their research interests at multi-layer social network extraction from activity data. The problem of multiple relations was for example investigated in [20] where the graphical, algebraic and sociometric notations for multiple relations are proposed. Wasserman and Faust proposed to use Image Matrices for Multiple Relations. However as authors emphasized interpreting each image (matrix a single relation) separately seems to be ad hoc. They suggest comparing pairs of images and investigating multi-relational patterns such as multiplexity or exchange. This solution does not solve the problem with networks where there can exist many of relation types.

Another approach was presented by Jung in [13] who proposed to combine social relationships with corresponding ontology and concept ties. In another approach, which is not semantically-driven, Kazienko *et al.* investigated Flickr photo sharing system and have distinguished eleven types of relationships between users [14]. A special type of social networks that allows the presentation of many different activities is called a multi-layered social network [3], [15]. It can be represented as a multi-graph [7], [20]. Overall, due to their complexity, such networks are more difficult to be analyzed than simple one-layered social networks and no established methods have been developed.

In this paper, we focus on developing a conceptual, generic model for multidimensional social network that enables to capture information about different types of activities and interactions between users as well as represent the dynamics of user's behaviour. The proposed model encompasses information not only about different relations but also the groups that exist within a given relation layer and in a specific time window.

3 Multidimensional Social Network: A Model

3.1 Multidimensional Model of a Social Network

The social network models should reflect real-world interactions between users with respect to all types of relationships existing between network's users accompanied by proper description, namely strength of relations and their dynamics. The representation of relations in the model should also allow gathering customized description of networks' and individuals' characteristics. The general idea behind the multidimensional model of social network endeavours to provide the framework allowing the description of entirety of social interactions existing between network actors.

Multidimensional model of social network presented in the paper is based on the basic profile of multidimensional and changing over time social networks. The foundation of each social network is a structure made up of individuals, which are tied by one or more specific types of interdependency, such as friendship, kinship, common interest, financial exchange, likes and dislike, etc. In order to represent such entities, the model assumes the representation of nodes and edges, where nodes represent individuals (social entities) and edges – interconnections between them. Obviously, as there exist many numbers of relationships types, thus edges represent distinct meanings of relations. Therefore, the model assumes that edges are grouped in separate semantic layers consisting of relations with the same meaning.

Social networks are not static structures and comprise relations that change over time. Thereby, the set of network actors may vary over time. The dynamics of relations and nodes needs its representation and is modelled by time windows – a set of static pictures (snapshots) representing the state of a network for a certain time interval.

The proposed model encompasses information not only about dynamics and different kinds of relations but also the groups that exist within a given relation layer and in a specific time window. It provides the opportunity to distinguish distinct sets of nodes with high density of internal edges and low density of edges between those sets.

Concluding, the general concept of the model considers three distinct dimensions of social networks: layer, time-window and group dimension, see Fig. 1. All the dimensions share the same set of nodes that corresponds to social entities: single human or groups of people.

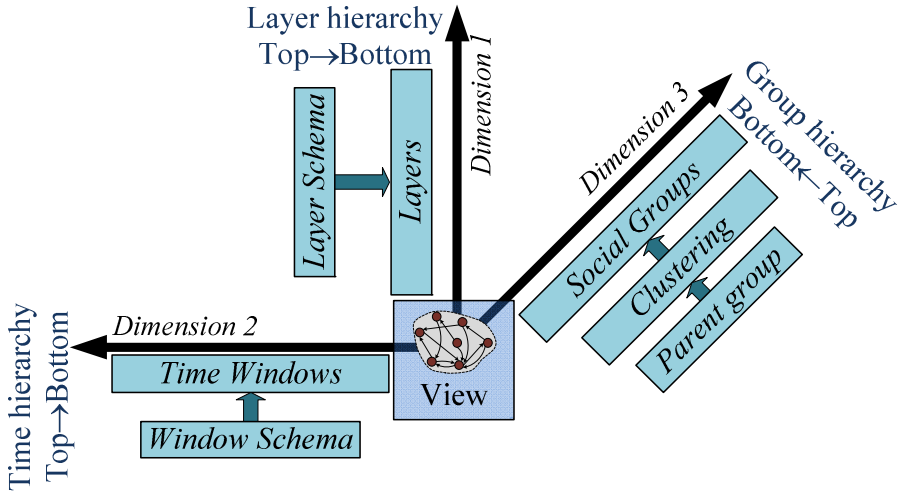


Fig. 1. Three dimensions with hierarchies in the multidimensional social network

3.2 Dimension 1: Layer Hierarchy

Layer dimension describes all the relationships between users of a system. The relations may represent direct communication between users via e-mail or phone. But they also may result from human activities in IT system, e.g. sharing and co-editing documents in business intranet.

In general, three categories of relations are distinguished: direct relation, pseudo-direct relation and indirect relation, see [14] for details. A relationship between the users may be directed, when it takes place from one user to another but not necessary the other way around, or undirected if a direction of the relation is not determined.

Besides, the relationships occurring between people have different nature. Going to the same school, shopping in the same e-magazines, being a friend of somebody, writing SMS to somebody, attending e-lectures are only few examples of the relation types. Based on the data available in a given system it is possible to extract all the types of relationships that occur between its users, defining in this way a set $\{R_1, R_2, \dots, R_n\}$, where $R_i = \{ \langle user_x, user_y \rangle \mid user_x, user_y \in Users, user_x \neq user_y \}$, $i=1, 2, \dots, n$ is a type of relation. Let IDU defines a finite set of users of one system and $L = \{l_1, l_2, \dots, l_n\}$ – a set of layers corresponding to the relations from the set $\{R_1, R_2, \dots, R_n\}$. Particular layers l_1, l_2, \dots, l_n consist of the same IDU set (nodes in graph representation) connected by relations (edges) of the types: R_1 in layer l_1 , R_2 in layer l_2 , and R_n in layer l_n respectively.

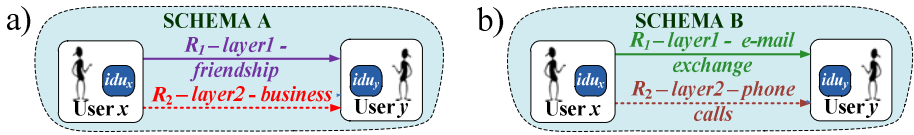


Fig. 2. Two dimensional social networks with layer dimensions created according to two different schemas: a) layer1 – friendship relationship, layer2 – business relationship, b) layer1 – e-mail exchange, layer2 – phone calls

Note that a single layer l_i represents simple social network including all users of the system connected to each other by relationship R_i , whereas two or more layers give multi-layered social network with the same set of nodes connected by more than one relationship.

Layer dimension can be created according to one or another Layer Schema (Fig. 2), e.g. Schema A: layer1 – *friendship*, layer2 – *business*, or Schema B: layer1 – *e-mail exchange*, layer2 – *phone calls*. Schemas form an additional upper level in the layer dimension hierarchy. As a result, the social network may be analyzed with respect to any simple (single relation) or different, complex (multi-relation) layer schemas.

3.3 Dimension 2: Time Hierarchy

Temporal analysis of social network is possible because of the existence of time-window dimension. A time-window is a period of time of a defined size. It may be a snapshot at a given time stamp, i.e. relation existing at that time, but also relations extracted for a given period, i.e. based on human activities within time-window, see Fig. 3.

Time-window limits social network analysis to those users (nodes) and relationships (edges) that have existed in a period defined by time-window size. Resulting social network may be simple (one-layered) or multi-layered. Missing information or changes prediction is then possible by comparing networks from successive time-windows, e.g. time-windows 1, 2 and 3 in Fig. 3.

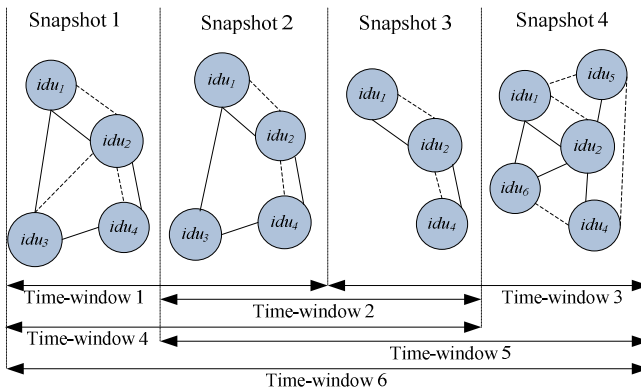


Fig. 3. Time window schemas and hierarchy

Basic problem of temporal SNA is time-window resolution. If time-window is too narrow structural parameters of social network are not correlated and the window itself introduces relatively big noise in parameter values. In turn, too wide time-window entails loss of information about temporal dependencies occurring between relations and nodes in social network. To solve this problem it is necessary to apply a method - like presented in [19] - that enables a choice of time-window optimal size.

Similarly to layer dimension, time-window dimension comprises time-windows with different sizes, moving windows, etc. that correspond to various Window Schemas, see Fig. 3.

3.4 Dimension 3: Group Hierarchy

Concept of group is not precisely defined in social network environment. In general, it is considered that group assembles similar peoples. Frequently, instead of definition we can find conditions (criteria) that should be fulfilled for the group to exist. Most of these conditions derive from an idea, that a group is a social community which members more often cooperate within the group than outside. So, in social network context, a group may be defined as a subset of users who are strongly connected with the other members of the group and loosely with members of other groups, see Group 1 and Group 2 in Fig. 4.

In the model of multilayered social network, group dimension is supposed to contain all the social groups possible to obtain in the clustering processes. However, different clustering algorithms may be applied. Clustering create the second level in the hierarchy of group dimension. In addition, a *Parent Group* concept is introduced. It is a virtual object - a root of a group hierarchy, which preserves information about inter-group relations used further in the aggregation process. A single social group may include a subset of social network users connected by a single relation or more than one relation in a given period of time. Thus it may be considered as multilayered structure in time-window. A group may also evolve over time. Its temporal changes and their dynamics give valuable information.

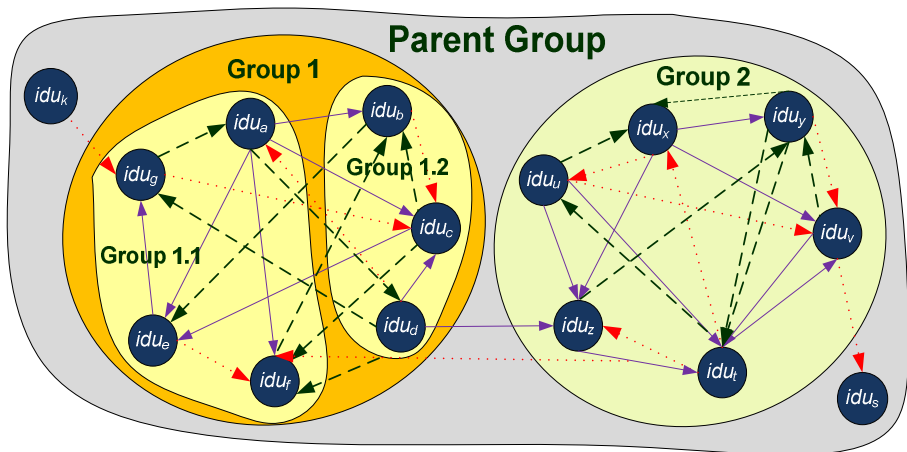


Fig. 4. Group hierarchy

3.5 Views – Dimensional Intersection

The dimensionality of the multidimensional model of a social network that is presented in the paper is utilized to conclude the state of the network providing its static picture. Therefore, the concept of view as a core of the model is introduced. The view is a sub-network consisting only of nodes and edges that belong to particular layer, time window and group. It means that the single view describes the state of the sub-network composed of nodes tied by edges representing the same type or relation between nodes, from the same time and that are in the same group of nodes. Note that, the concept of the model of a multidimensional social network may be compared with principal assumptions of logical architecture of data warehouses.

4 Multidimensional Social Network: Analysis

4.1 Aggregations by Dimensions

The proposed concept of atomic insight in the sub-network stated by views does not allow performing queries to more sophisticated structures composed of several views. There is a strong expectation to provide the possibilities to operate on multiple views in order to consider not only a single view but more compounded patterns from the entire network. Therefore, some aggregation operators working on dimensions are required. Aggregations should offer ability to analyze such sub-network structures like accumulated network activity from particular layers, time-windows or groups. For instance, one can perform analysis of the network, considering activity from selected time-windows aggregating a given single hour of the day for all the days in the week only.

What is more, views can be aggregated by one, two or even all three dimensions at the same time. The aggregation creates a new social network object composed of nodes and edges from the considered views but with recalculated relation strengths. This recalculation is accomplished by taking into account only those relationships that occur in the selected views.

There might be considered several approaches of views' aggregation. Among others, typically, the relation strengths may be aggregated by:

- sum of relation strengths,
- mean of relation strengths,
- weighted sum of relation strengths,
- weighted mean of relation strengths.

All above mentioned aggregations are performed on edges existing between the same pair of nodes but in distinct views. Additionally, another aggregation for set of nodes appearing in distinct views may be performed by union of sets of nodes. Aggregation operations may consider additional profile of relations – timeliness. As a result, older relations can be treated as less significant in strength calculation.

4.2 Usability of the Model

In Figure 4, the simple system, which utilizes multidimensional social network described above to analyse a large social networking system and big organizations, is presented.

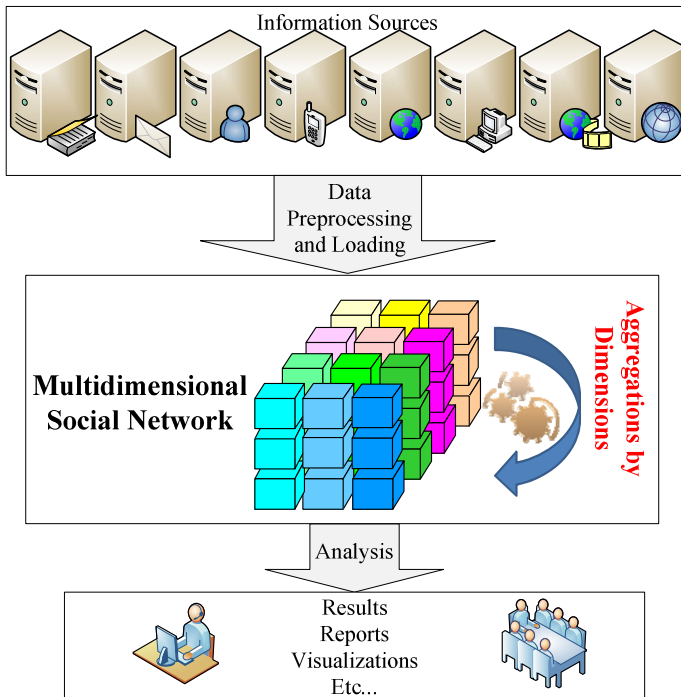


Fig. 4. Usage of Multidimensional social network

Systems, where people are linked by many different relationship types like in complex social networking sites (e.g. Facebook), where people are connected as friends, via common groups, “like it”, etc. or in regular companies: department colleagues, best friends, colleagues from the company trip, etc., can be analysed using layer dimension. Multidimensionality provides a chance to analyse each layer separately and at the same time investigate different aggregations of layer dimension. For example, let us consider a network consisting of six layers, three from the real world: family ties, work colleagues and gym friends and three from the virtual world, i.e. friends from Facebook and fiends from the MMORPG game and friends from Stargate forum. Now, one has three different possibilities to analyse such network: (i) analyse each layer separately, (ii) aggregate layers from the real world and compare it to the virtual world layers aggregation, and finally, (iii) aggregate all layers together.

Time dimension provides possibility to investigate the network evolution and its dynamics. For example, the analysis how users neighbourhoods change when one of the neighbours leave the network and how it affects the network in longer period of time, how group leaders (e.g. project team leaders) change over time, or how changes on one layer affect the other layers.

Finally, the group dimension. It allows studying groups existing within the social network. Using multidimensionality not only the usual social groups can be analysed (friend family, school, work, etc.) but also groups created upon various member

features like gender, age, location etc. Moreover, the model allows to compare the results of different community extraction methods, e.g. by means of social community extraction or typical data mining clustering.

To conclude, the multidimensional social network enables to analyse all three dimensions at the same time, e.g. how interaction on different layers of two social groups changes over time. Moreover, any measure can be calculated separately for view, layer, window, group, or any aggregation of the above and next, compared to each other to find the specific characteristics of each dimension, or the network can be analysed as a whole. Thus, the network multidimensionality opens new possibilities for social network analysis.

5 Conclusions and Future Work

Large amount of data about users and their activities has created a need for multidimensional analysis as one can observe many types of relationships that additionally change in time.

The proposed model of multidimensional social network enables a comprehensive and detailed description of social interactions among individuals as well as the profiles of users and their relationships. Such model can be used to investigate the relationships based on their features and also to analyse different ways of communication in the context of social science. For example one can analyse which modes of communication are jointly used or whether one mode of communication supports another one. Moreover, the model that includes the information about different types of connections between users facilitates development of new and redefinition of existing characteristic features describing users and networks.

The proposed model also includes the information about the communities that exist at each relation layer. This provides knowledge about within which relation types (activities) groups are more likely to form.

Both different relation layers and communities existing within these layers can be investigated in different time windows. It means that their evolution and dynamics can be examined.

To sum up the development of a formal model for multidimensional social network is a novel and interesting idea that enables to investigate each complex social network from different perspectives.

Acknowledgments. This work was supported by The Polish Ministry of Science and Higher Education, the development project, 2009-11, the research project, 2010-13, Fellowship co-financed by European Union within European Social Fund and the project of the City of Wrocław, entitled — "Green Transfer" – academia-to-business knowledge transfer project co-financed by the European Union under the European Social Fund, under the Operational Programme Human Capital (OP HC): sub-measure 8.2.1.

References

1. Agarwal, N., Galan, M.H., Liu, H., Subramanya, S.: WisColl: Collective Wisdom based Blog Clustering. *Information Sciences* 180(1), 39–61 (2010)

2. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech.*, 10008 (2008)
3. Bródka, P., Musiał, K., Kazienko, P.: A method for group extraction in complex social networks. In: Lytras, M.D., Ordóñez De Pablos, P., Ziderman, A., Roulstone, A., Maurer, H., Imber, J.B. (eds.) WSKS 2010. CCIS, vol. 111, pp. 238–247. Springer, Heidelberg (2010)
4. Cheng, X., Dale, C., Liu, J.: Statistics and social networking of YouTube videos. In: Proc. the 16th International Workshop on Quality of Service, pp. 229–238. IEEE, Los Alamitos (2008)
5. Chiu, P.Y., Cheung, C.M.K., Lee, M.K.O.: Online Social Networks: Why Do "We" Use Facebook? In: The First World Summit on the Knowledge Society. *Communications in Computer and Information Science*, vol. 19, pp. 67–74. Springer, Heidelberg (2008)
6. Ellison, N.B., Steinfield, C., Lampe, C.: The benefits of Facebook "friends:" Social capital and college students' use of online social network sites. *J. of Computer-Mediated Communication* 12(4),art.1 (2007), <http://jcmc.indiana.edu/vol12/issue4/ellison.html>
7. Flament, C.: *Application of graph Theory to Group Structure*. Prentice-Hall, Englewood Cliffs (1963)
8. Garton, L., Haythornthwaite, C., Wellman, B.: Studying Online Social Networks. *Journal of Computer-Mediated Communication* 3(1), 75–105 (1997)
9. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. *The National Academy of Sciences, USA* 99(12), 7821–7826 (2002)
10. Golbeck, J., Hendler, J.: FilmTrust: movie recommendations using trust in web-based social networks. In: *IEEE Conference Proceedings on Consumer Communications and Networking Conference*, vol. 1, pp. 282–286 (2006)
11. Hanneman, R., Riddle, M.: *Introduction to social network methods*. Online textbook. University of California, Riverside (2005), <http://faculty.ucr.edu/~hanneman/nettext/>
12. Huberman, B., Romero, D., Wu, F.: Social networks that matter: Twitter under the microscope. *First Monday*, 1–5 (2009)
13. Jung, J.J.: Query transformation based on semantic centrality in semantic social network. *Journal of Universal Computer Science* 14(7), 1031–1047 (2008)
14. Kazienko, P., Musiał, K., Kajdanowicz, T.: Multidimensional Social Network and Its Application to the Social Recommender System. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans* 41(4) (2011) (in press)
15. Kazienko, P., Musiał, K., Kajdanowicz, T.: Profile of the Social Network in Photo Sharing Systems. In: *AMCIS 2008, Association for Information Systems, AIS* (2008)
16. Kazienko, P., Musiał, K., Zgrzywa, A.: Evaluation of Node Position Based on Email Communication. *Control and Cybernetics* 38(1), 67–86 (2009)
17. Kazienko, P., Ruta, D., Bródka, P.: The Impact of Customer Churn on Social Value Dynamics. *Int. J. of Virtual Communities and Social Networking* 1(3), 60–72 (2009)
18. Scott, J.: *Social Network Analysis: A Handbook*. SAGE Publications, London (2000)
19. Sulo, R., Berger-Wolf, T., Grossman, R.: Meaningful Selection of Temporal Resolution for Dynamic Networks. In: *MLG 2010. ACM, New York* (2010)
20. Wasserman, S., Faust, K.: *Social network analysis: Methods and applications*. Cambridge University Press, New York (1994)
21. Watts, D.J., Strogatz, S.: Collective dynamics of 'small-world' networks. *Nature* 393, 440–444 (1998)
22. Wellman, B., Salaff, J., Dimitrova, D., Garton, L., Gulia, M., Haythornthwaite, C.: Computer Networks as Social Networks: Collaborative Work, Telework, and Virtual Community. *Annual Review of Sociology* 22(1), 213–238 (1996)

Modelling and Simulation of an Infection Disease in Social Networks

Rafał Kasprzyk, Andrzej Najgebauer, and Dariusz Pierzchała

Military University of Technology, Cybernetics Faculty,
Gen. Sylwestra Kaliskiego Str. 2, 00-908 Warsaw, Poland
{rkasprzyk, anajgebauer, dpierzchala}@wat.edu.pl

Abstract. The paper focuses its attention on a software project that takes advantage of pioneering sociological theories, graph & network theory, and the state-of-the-art in software technologies. Its very purpose, of particularly high importance nowadays, is to counter infectious diseases. The paper refers to research of *Complex Networks* displaying the, so called, *Scale Free* and *Small World* features, which make them accurate models of Social Networks.

Keywords: social and complex networks, epidemic modelling and simulation.

1 Basic Definition and Notation

Formally, a graph is a vector $G = \langle V, E, P \rangle$ where: V is a set of vertices, E is a set of edges, and P is an incidence relationship, i.e. $P \subset V \times E \times V$. The degree k_i of a vertex v_i is the number of edges originating from or ending in vertex v_i . The shortest path d_{ij} from v_i to v_j is the shortest sequence of alternating vertices and edges, starting in vertex v_i and ending in vertex v_j . The length of a path is defined as the number of edges in it. Networks very often are represented by a adjacency matrix A , which is an $n \times n$ matrix, where n is the number of vertices in the network, $n = |V|$. Element of adjacency matrix $A_{ij} = 1$, if there is an edge between vertices i and j , and 0 otherwise.

Networks are commonly modelled with either simple or directed graphs where the set of vertices represents objects, and joins (arcs, edges) between two vertices exist if the corresponding objects are related due to some relationship. In some cases the use of graph does not provide a complete description of the real-world systems. For instance, if contacts in Social Networks are represented as a simple graph, we only know whether individuals are connected, but we cannot model strength of that connection. For now, however, we will use only formal graph definition.

2 Epidemic Modelling in Social Networks

The network of possible contacts between individuals describes which individuals can infect which. We explore epidemic spreading in Social Networks modelled by Complex Networks [4]. One of the most known mathematical models of Social Networks generators was a random graph [1]. Assuming equally probable and independent random connections between any two vertices in initially not connected graph, they derived a model with a highly unrealistic social network topology.

Apparently, Complex Networks have *Scale Free* [3] and *Small World* [2] features. These features, which appear to boost efficiency in communication networks, at the same time quicken the spreading of many diseases. A *Small World* network is a type of graph in which most nodes are not neighbors of one another, but most of them can be reached from any other with a small number of steps. The *Scale Free* feature pertains to a network in which most of people have relatively small amount of contacts, but there are some individuals that have huge amount of contacts. These individuals are called “*super-spreaders*”, because they can spread diseases very fast. If such individual gets infected and in turn infects a portion (or perhaps all) of his numerous neighbors, that causes a sudden increase in the count of sick people. The application uses a few centrality measures [6][11] that help in finding the critical elements (e.g. degree centrality, closeness centrality or betweenness centrality) and finally suggests who should be immunized [14].

It is important to notice that we can determine the dynamic of epidemics if we know the network of possible contacts between people. Thus, the knowledge of such network topology makes it possible to simulate spreading of contagious diseases and to counteract them effectively. The system is a novel attempt in countering spreading of diseases, the first one to incorporate the knowledge stated above. Today there is no similar solution, and in most cases epidemiologists still choose people to vaccinate at random or decide to vaccinate the whole population if they have enough vaccines. Unfortunately, the most frequent situation is that we do not have enough vaccines to treat the whole population. And random immunization is almost useless, because it gives a very small chance of separating a social network into independent components – it is characteristic for *Scale Free* networks that they remain connected even after up to 80% of their nodes are removed (in our case: immunized or isolated) [10]. This suggests a simple solution: to immunize the “*super-spreaders*” first, which will slow or stop the spread. Nevertheless, this solution is very often impossible because in most cases the knowledge of network topology is uncertain and incomplete.

The described system proposes a number of solutions of gathering and using the knowledge of social networks features. It also gives a tool to generate synthetic social networks with the same statistical properties as real networks [1][2][3][5][12].

3 Social Networks Generators

In 1960, Erdős and Rényi [1] assumed equally probable and independent random connections made between any pair of vertices and derived a model that suffered unrealistic topology. Although their model was not very useful for modelling real life social network, they proved a number of interesting result about random graphs.

Identifying and measuring properties of Social Networks is the first step towards understanding their topology, structure and dynamics. The next step is to develop a mathematical model, which typically takes a form of an algorithm for generating networks with the same statistical properties. Apparently, networks derived from real data (most often spontaneously growing) have “*six degree of separation*”, power law degree distributions, hubs occurring, tendency to form clusters, etc. Two very interesting models capture these feature, have been introduced recently.

First, Watts and Strogatz in 1998 [2] deal with mentioned features by a strategy that seems perfectly obvious once someone else has thought of it. They interpolate between two known models. They begin with a regular lattice, such as a ring, and then introduce randomness by ‘rewiring’ some of the edges. The process of rewiring affects not only the average path’s length but also the clustering coefficient. Both of them decrease as probability of rewiring increases. The striking features of this procedure is that for relatively wide range of rewiring probabilities the average path length is already low while clustering coefficient remains high. It is called *Small World* model, or more precisely: Beta-model of *Small World* network. Next to Beta-model there exists also Alfa-model of *Small World* network [5]. What is surprising is not that real Social Networks are *Small World* but that people are able to find the shortest path between each other so easily.

Second, Barabasi and Albert in 1999 [3] introduced their model of networks as a result of two main assumption: constant growth and preferential attachment. They expressed the degree sequence – the count of vertices with the same degree (number of adjacent edges) for all degree values found in the network. The network grows gradually, and when a new node is added, it creates links to the existing nodes with probability proportional to their current connectivity. This way, highly connected individuals receive more new links than not so connected ones, and also, ‘old’ nodes are more connected than ‘young’ ones. It is called *Scale Free* model. The process of *Scale Free* networks generation has many extensions and modifications [12].

4 Centrality Measures

We start the analysis of epidemic spreading by introducing centrality measures [6][11], which are the most fundamental and most frequently used measures of network structure. The central vertices in Complex Networks are of particular interest because they might play the role of organization hubs. Centrality measures addresses the question of “Who (what) is the most important or central person (node) in given social network?”. No single measure of center is suitable for all application. Based on the defined centrality measures, we show how to discover the critical elements of any network, the so-called “*super-spreaders*” of a disease. When a vaccination for a disease exists, immunizing certain individuals may be the most efficient way to prevent loss of time and funds due to the disease. Obviously, immunization of the entire population will eradicate the disease entirely, but this is not always possible, or may involve high cost and effort. Therefore, the choice of individuals to immunize is an important step in the immunization process, and may increase the efficiency of the immunization strategy. We considered five most important centrality measures:

Degree Centrality

The simplest of centrality measures is degree centrality, also called simply degree. The degree centrality measure gives the highest score of influence to the vertex with the largest number of direct neighbors. This agrees with the intuitive way to estimate someone’s influence from the size of his immediate environment: $k_i = \sum_{j=1}^n A_{ij}$. The degree centrality is traditionally defined analogically to the degree of a vertex, normalized with the maximum number of neighbors that this vertex could have. Thus, in a network of n vertices, the degree centrality of vertex v_i , is defined as:

$$center_i^{Degree} = \frac{k_i}{n-1} \quad (1)$$

Radius Centrality

If we need to find influential nodes in an area modelled by a network it is quite natural to use the radius centrality measures, which chooses the vertex using pessimist's criterion. The vertex with the smallest value of shortest longest path is the most central node. So if we need to find the most influential node for the most remote nodes it is quite natural and easy to use this measure. The radius centrality of vertex v_i , can be defined as:

$$center_i^{Radius} = \frac{1}{max d_{ij}} \quad (2)$$

Closeness Centrality

This notion of centrality focuses on the idea of communication between different vertices. The vertex which is 'closer' to all vertices gets the highest score. In effect, this measure indicates which one of two vertices needs fewer steps in order to communicate with some other vertex. Because this measure is defined as 'closeness', the inverse of the mean distance from a vertex to all others is used.

$$center_i^{Closeness} = \left[\frac{\sum_{j=1}^n d_{ij}}{n-1} \right]^{-1} = \frac{n-1}{\sum_{j=1}^n d_{ij}} \quad (3)$$

Betweenness Centrality

This measure assumes that the greater number of paths in which a vertex participates, the higher the importance of this vertex is for the network. Betweenness centrality refines the concept of communications, introduced in closeness centrality.

Informally, betweenness centrality of a vertex can be defined as the percent of shortest paths connecting any two vertices that pass through that vertex. If $p_{lk}(i)$ is the set of all shortest paths between vertices v_l and v_k passing through vertex v_i and p_{lk} is the set of all shortest paths between vertices v_l and v_k then:

$$center_i^{Betweenness} = \frac{2 \sum_{l < k} \frac{p_{lk}(i)}{p_{lk}}}{(n-2)(n-1)} \quad (4)$$

This definition of centrality explores the ability of a vertex to be 'irreplaceable' in the communication of two random vertices. It is of particular interest in the study of network immunization, because at any given time the removal of the vertex with the highest betweenness seems to cause maximum damage to the network in terms of its connectivity and mean distance.

Eigenvector Centrality

Where degree centrality gives a simple count of the number of connection that a vertex has, eigenvector centrality acknowledges that not all connections are equal. In general, connections to people who are themselves influential will grant a person more influence than connections to less important people. If we denote the centrality of vertex v_i by e_i , then we can allow for this effect by making e_i proportional to the average of the centralities of the v_i 's network neighbors.

$$e_i = \frac{1}{\lambda} \sum_{j=1}^n A_{ij} e_j \Rightarrow \vec{e} = \frac{1}{\lambda} A \vec{e} \Rightarrow A \vec{e} = \lambda \vec{e} \quad (5)$$

So we have $A\vec{e} - \lambda I \vec{e} = 0$ and the λ value we can calculate using $\det(A - \lambda I) = 0$. Hence, we see that \vec{e} is an eigenvector and λ – an eigenvalue of the adjacency matrix. We wish the centralities to be non-negative so it can be shown that λ must be the largest eigenvalue of the adjacency matrix and \vec{e} the corresponding eigenvector.

5 Connection Efficiency

To evaluate how well a network is connected before and after the removal of a set of nodes we use the global connection efficiency (GCE)[10]. The connection efficiency between vertex v_i and v_j is inversely proportional to the shortest distance:

$$connection_{ij}^{efficiency} = \frac{1}{d_{ij}} \quad (6)$$

When there is no path in the graph between vertex v_i and v_j we have $d_{ij} = infinity$ and consequently connection efficiency is equal zero. The global connection efficiency is defined as the average of the connection efficiency over all pairs of nodes.

$$GCE = \frac{2}{n(n-1)} \sum_{i < j} \frac{1}{d_{ij}} \quad (7)$$

Unlike the average path length, the global connection efficiency is a well-defined quantity also in the case of non-connected graphs.

6 Vaccination Strategies

Random immunization of social networks is almost useless because *Scale Free* networks remain connected even after up to 80% of their all nodes removed (immunized or isolated) [10]. This would mean that under random vaccination, almost the whole population must be vaccinated to prevent the disease's spread. However, a clever attack (targeted vaccination) aimed at "*super-spreader*" will disintegrate the network rapidly. So, if we know social network topology we can use centrality measures to identify most important nodes and then vaccinate only those with the highest score to stop the disease.

It might be harder to come up with a clever strategy when we do not know the topology of social network. The question here is: how to identify and/or find the "*super-spreaders*" if we are not able to calculate values of centrality measures? It can be accomplished with a simple modification of random vaccination based on a new concept introduced in [13] [14] with few modification. According to our computer simulation results the new vaccination strategy it is much more effective, also in the case when our knowledge of the network topology is uncertain and incomplete.

The idea is to randomly choose, say, 20% of the individuals and ask them to fill out our special questionnaires. One of the most important question in all forms for any disease is to name at least one acquaintance/friend/partner/colleague etc., and then vaccinate those identified individuals (vaccinate the neighbors). Potential "*super-spreaders*" have such a large number of contacts that they are very likely to be named at least once. On the other hand, the "*super-spreaders*" are so few in number that the random sample of individuals is unlikely to include many of them. Using this

vaccination strategy, a disease can be stopped by vaccinating less than 20% of individuals. If a larger sample is polled, or those named twice are vaccinated the total number of vaccinations required can be even lower. This basic method can be modified in many ways and be adapted to different *Scale Free* networks [7] and a specific disease or virus based on simulation result generated in CARE².

7 System Architecture Overview

The CARE² software is a distributed system consisting of a server and different types of clients. At the moment there are two types of clients, i.e. a web client accessible through web browser and a mobile platform client. Other developers could also access web services and design their completely new interface and/or analyses. The system was developed using *SOA Architecture* with *Web Services* and cloud computing approach to massive calculation. All calculations are executing “in cloud” i.e. networks generation, algorithms or simulation. The system has been implemented on *Microsoft .NET 3.5* platform using *Visual Studio 2010 beta2* and *Expression Blend 3* (+preview for .NET4 extension), and *Azure Platform* as Microsoft implementation of cloud computing idea. The web user interface uses ASP.NET technology with AJAX (*Asynchronous JavaScript and XML*) and *Microsoft Silverlight 4.0* solution. For “geo-contextual” services *Bing Maps* is used. Mobile clients use *J2ME* platform and *LWUIT* library. From *Microsoft Research* we apply *QuickGraph 3.0* - Generic Graph Data Structures and Algorithms for .NET platform and *SMSServerToolkit*.

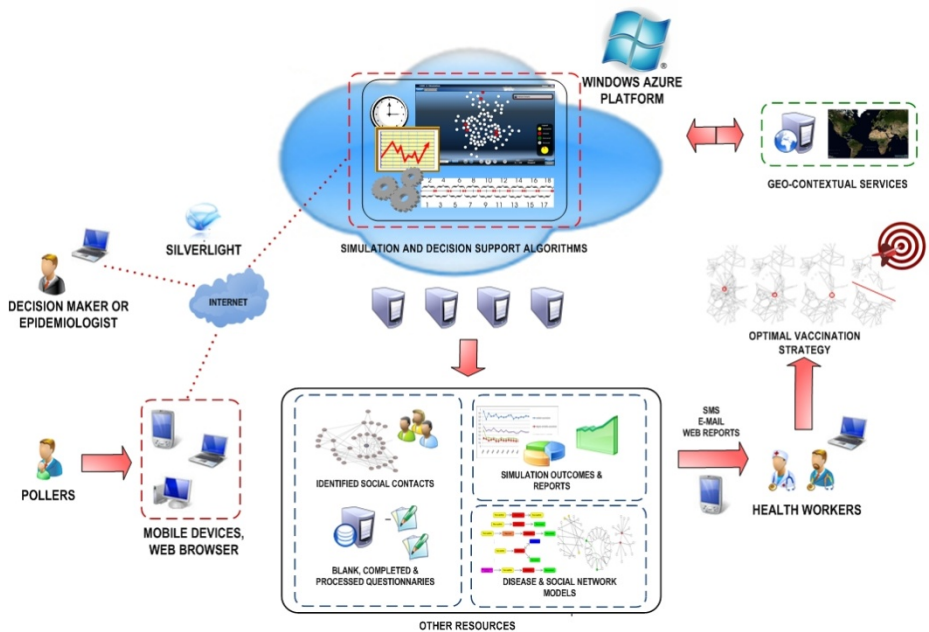


Fig. 1. CARE² architecture overview

The system allows users to:

- model any kind of disease based on epidemiological knowledge;
- model and generate social networks using Complex Network theory;
- simulate and visualize how the epidemic will spread in a given population;
- assess the expected outcomes of different simulation scenarios.
- identify “*super-spreaders*” and come up with efficient vaccination strategies;
- build special polls running on PDA to help discover network topology.

CARE2 takes advantages of experiences acquired thanks to CARE [16] developing two years ago. Even so CARE2 is entirely new solution with new functionality implemented concerning for the quality of service. While CARE was an initialization of a framework to test some concepts, CARE2 is complete mature work.

7.1 Disease Modelling

Using State Machine approach we can model any kind of disease based on knowledge from the field of epidemiology. We allow to build the models of diseases with any state in the editor we have proposed. It means that every disease consists of a few states (e.g. susceptible, infected, carrier, immunized, dead etc.) which can be assigned to each individual and the system allows state to change as a result of social interactions (contacts). So underlying network topology is crucial problem in our simulator. For simulation we need at least two states S (*Susceptible*) and I (*Infected*). For realistic scenarios we make possible to define the model of disease with many more states and transitions between them. In the classical theory of infectious diseases, the most extensively studies epidemic models are the SIR (*Susceptible-Infected-Removed*) and SIS (*Susceptible-Infected-Susceptible*) model.

We allow to build the models of diseases with any state in the editor we have proposed. We are able to define some essential parameter like: transition probability between states, minimum/maximum time that an individual spends in each state, maximum number of neighbours that can be infected by individual in a given time period and much more.

7.2 Gathering Raw Distributed Data

The crucial step in fighting against a disease is to get information about social network subject to that disease. The CARE² software allow user to load or generate the right network. The system uses *graphML* network format to keep networks in external sources. Using proposed generators we obtain synthetic networks but with the same statistical properties as real social networks. The algorithms generate networks that are *random graphs*, *Small World* networks, *Scale Free* networks or modifications thereof. The CARE² software contains also special module to get information about real social networks. It helps building questionnaires based on sociological knowledge to help discover network topology. Questionnaires design in this way are deployed on mobile devices to gather social data in the field.

In order to verify and validate CARE² simulation outcomes we implemented subsystem of CARE² called SARNA dedicated to gather and monitoring all necessary data related to infectious diseases spreading across Poland. SARNA was organized as a collection of geographically-distributed software components (based upon SOA idea). Some of components were legacy software and thus whole system was perceived as heterogeneous, from both hardware and software point of view.

Let's take into consideration an influenza virus (flu) which concerns almost each country of the world. In the past winter seasons, its new strain called swine influenza (A H1N1) infected so many people that the problem of lack of hospital beds appeared. In Poland, in the high risk of infection, the Government Safety Centre (Polish acronym: RCB) had to carry out tasks of daily monitoring of influenza and influenza-like illness in the country.

Data entered into authorized questionnaires were divided into two groups: summary information for the period of 24 hours (the number of persons admitted for hospitalization and discharged from a hospital, the number of deaths of people with infectious respiratory disease) and current information at 8.00 a.m. (the number of hospitalized persons and hospitalized persons, who require ventilatory support as well as the number of free beds with possibilities of assist breathing with the respirator). In both groups, the data have been divided into two age categories: children (aged up to 14 years) and adults (persons aged 15 years or more). Finally, gathering of relevant data enabled creating reports, forecasting effects, estimating trends, simulation of crisis processes and optimising decisions.

Before 2009, the integrated epidemiological and virological surveillance of influenza (SENTINEL) was the sole way to obtain information related to flu situation. It operates in 26 European countries according to the EISS and WHO guidelines. A select group of "sentinel providers" (frequently GP) report the total number of patients and the total number of patients with complaints of illness consistent with flu only once a week. This way gives an aggregated view at a type of virus but, for instance, no information about the frequency of its occurrence. The proposed monitoring subsystem has enabled increasing sensitivity for detection of epidemics due to collecting information every day. Moreover, it has led to shorter response times to critical changes. During the 2009/2010 winter, a process of data acquisition for building of a knowledge base was supported by 600 poviats hospitals.

After the winter season, we obtained the possibility of validation and testing of the adequacy of the system. The VV&A method uses data stored in both the systems: SENTINEL and the proposed subsystem. That quantitative approach uses indicators describing the dynamics of time series created from the data collected and divided into the following categories:

- the number of patients admitted to poviats hospitals with a positive flu test,
- the number of deaths associated with both the seasonal flu and H1N1.

The data set is the 3-month snapshot: from 23 November 2009 to 31 March 2010. The presented data [15] were obtained from reports provided by the State Sanitary Inspectorate (www.pzh.gov.pl/epimeld) and the Government Safety Centre (www.rcb.gov.pl). The conclusion is that the estimated trends are very similar and the subsystem SARNA could be used to calibrate models of infectious diseases as well as models of social networks used in CARE².

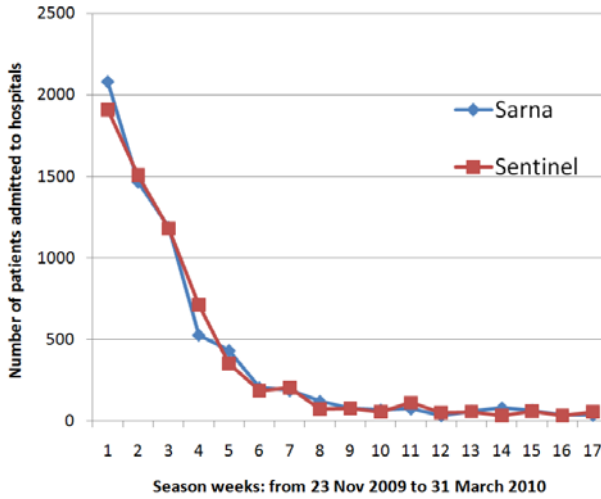


Fig. 2. Hospitalized flu patients

7.3 Simulation and Reports

Thanks to this module user can simulate and visualize how the epidemic will spread in a given population. the user can modify several parameters like: number of simulation steps and initial conditions for the simulation – which include the number and pattern (randomly, by a chosen centrality measure) of infected individuals and vaccination strategy (random, vaccinate thy neighbour, by chosen centrality measure).

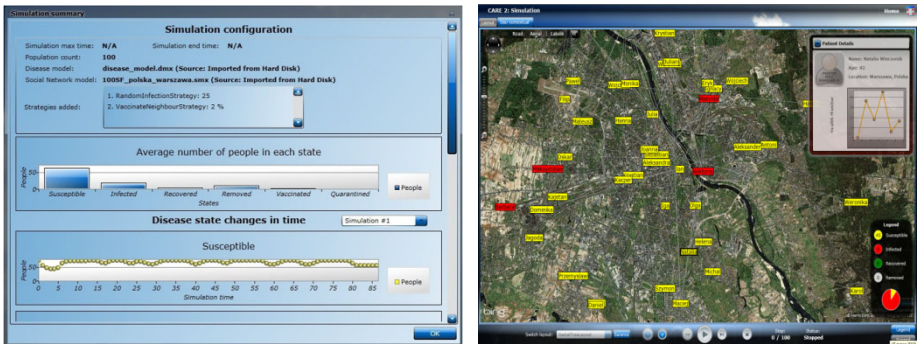


Fig. 3. The “report chart” and “geo-contextual” visualization of the simulation

The system proposes two ways of information visualization. The first way is called “layout” and help user to manipulate networks and some parameters of simulation. The alternative way is „geo-contextual” one which allows to visualize networks on the world map.

The system estimates the expected outcomes of different simulation scenarios and generate detailed reports. The user can assess the results and the effectiveness of the chosen vaccination strategy. A report chart is created on the basis of the simulation. The x-axis represents simulation steps and the y-axis represents a count of individuals in each state in appropriate steps.

7 Conclusions

Based on the defined centrality measures, we have shown how to discover the critical elements of any network. The identification and then vaccination of such critical individuals in a given network should be the first concern in order to reduce the consequence of an epidemics.

The CARE² software has enormous practical potential in regions, where there are not enough medicines or time to treat all those at risk. It could be also used by crisis management centres and epidemiology centres in the whole world to fight against any kind of infectious diseases.

Acknowledgments. This work was partially supported by the research project GD-651/2011/WCY of Cybernetic Faculty at Military University of Technology.

References

1. Erdős, P., Rényi, A.: On random graphs, *Publ. Math. Debrecen.* 6, 290–297 (1959)
2. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world” networks. *Nature* 393, 440–442 (1998)
3. Barabasi, A.L., Reka, A.: Emergence of Scaling in Random Networks. *Science* 286, 509–512 (1999)
4. Barabasi, A.L., Reka, A.: Statistical mechanics of complex networks. *Review of Modern Physics* 74, 47–97 (2002)
5. Watts, D.J.: *Small Worlds: The Dynamics of Networks Between Order and Randomness.* Princeton University Press, Princeton (1999)
6. Harary, F., Hage, P.: Eccentricity and centrality in networks. *Social Networks* 17, 57–63 (1995)
7. Kasprzyk, R.: The vaccination against epidemic spreading in Complex Networks, *Biuletyn ISI, Nr 3/2009, Warszawa, vol.(3) (2009) ISSN 1508-4183*
8. Brandes, U., Kenis, P., Raab, J.: Explanation Through Network Visualization. *Methodology* 2(1), 16–23 (2006)
9. Pastor-Satorras, R., Vespignani, A.: Epidemic Spreading in Scale-Free Networks. *PRL* 86(14), 3200 (2001)
10. Crucitti, P., Latora, V., Marchiori, M., Rapisarda, A.: Error and attack tolerance of complex networks. *Physica A* 340, 388–394 (2004)
11. Wuchty, S., Stadler, P.F.: Centers of complex networks. *Journal of Theoretical Biology* 222, 45–53 (2003)
12. Barabási, A.L., Albert, R.: Topology of Evolving Networks: Local Events and Universality. *PRL* 85(24), 5234 (2000)

13. Cohen, R., Havlin, S., ben-Avraham, D.: Efficient Immunization Strategies for Computer Networks and Population. PRL 24, 247901–247901 (2003)
14. Madar, N., Kalisky, T., Cohen, R., ben-Avraham, D., Havlin, S.: Immunization and epidemic dynamics in complex networks. Eur. Phys. J. B 38, 269–276 (2004)
15. Najgebauer, A., Pierzchała, D., Kasprzyk, R.: A distributed multi-level system for monitoring and simulation of epidemics. In: Brebbia, C.A. (ed.) Risk Analysis VII and Brownfields V, pp. 583-596. WITPress (2010) ISBN 978-1-84564-472-7
16. Kasprzyk, R., Najgebauer, A., Pierzchała, D.: Creative Application to Remedy Epidemics. In: Brebbia, C.A. (ed.) Risk Analysis VII and Brownfields V, pp. 545–562. WIT Press (2010) ISBN: 978-1-84564-472-7

Distributed Military Simulation Augmented by Computational Collective Intelligence

Dariusz Pierzchała, Michał Dyk, and Adam Szydłowski

Military University of Technology, Faculty of Cybernetics,
Gen. Sylwestra Kaliskiego Str. 2, 00-908 Warsaw, Poland
Ph.: +48226839504; Fax: +48226837858

dpierzchala@wat.edu.pl, {michal.dyk,aresius0}@gmail.com

Abstract. Contemporary peacekeeping and peace-making military operations, being conducted in the settings of an urban environment, need effective and adequate distributed simulation systems in order to rapidly and properly train commanders and their subordinate soldiers. The main requirements are to adequately model typical and unique behaviours, different habits and minds of the civilian groups of people as well as to constructively simulate a battlefield at different levels of resolution (soldiers acting separately or in groups). The paper considers a software which combines a high resolution simulation of collective intelligence (in VBS2) with constructive simulator, using HLA standard. Moreover, the algorithms of behaviour for simulation of soldiers and civilians, resulting from new types of dangers emerging during the asymmetric urban operations, have been defined. Finally, quality measures and methods of calculation to quantitative evaluation of proposed approach are proposed.

Keywords: collective intelligence, distributed military simulation, HLA.

1 Introduction

Contemporary military missions are focused on Peacekeeping, Peace Support and Peace-making Operations which are strictly connected not only with human relief activities but frequently with actions against civilian groups, especially in the settings of an urban environment, as well. These types of operations have implicated significant changes in both a decision making and a training as well as planning process. The urgent need of effective and adequate simulation systems has its roots in a rapid deployment of forces in hostile territories where they are faced with asymmetric threats. Consequently, soldiers are forced to conduct war amongst the people and against the people. It may request the deepest knowledge about their typical and unique behaviours, different habits and minds. Real military systems are very complex, that's why no single, monolithic simulation can satisfy the needs of all users. In order to widely prepare to an operation in a shorter time, the current state-of-the-art behaviour models (for new types of dangers coming from the asymmetric urban military operations) and data exchange mechanisms implemented in distributed simulation are necessary. The presented approach may have a significant impact on a mission success. It might be achieved using distributed software systems for a high

resolution simulation of collective intelligence (such as civilian group decision making and coordination of collective actions) combined with constructive stochastic battlefield simulation. Finally, quality measures and methods of calculation in order to quantitative evaluation of the proposed approach should be defined.

A distribution of conflict model components is natural and results from the real system's objects distribution. An integrated simulation environment for mission planning, training and rehearsal for both staff and field personnel on a tactical level can be realized and executed in many various ways. Current state of information technology enables the construction of software environment with useful methodologies and software – the best known standards in a simulation are: DIS – Distributed Interactive Simulation and HLA – High Level Architecture [4]. There are many existing or legacy simulators that from one side completely cover almost all real problems and from other side some of them are already not adequate and obsolete. The important issue is that simulators (e.g. CBS, JTLS, ModSAF, Zlocien [1], [3]) have different types of combat models – it implicates different representations of a battlefield actions. In a case of tactical level it means simulating small units in high density environment (usually using virtual simulation). For modelling of larger military forces, and their specific tasks, a constructive simulation is applied.

In the case study we have focused on the two systems: SSWSO Zlocien (created at Cybernetics Faculty, Military University of Technology with paper authors' contribution in a simulation module), which has been put into practice in Polish Army, and Virtual Battlespace 2 (VBS2) developed by Bohemia Interactive Studio and Bohemia Interactive Australia. The first one is a constructive stochastic simulator for a Computer Assisted Exercises (CAX) at “platoon to brigade” levels. The VBS2 is an interactive environment for a military training of a single person or a small group of soldiers. These systems have brought a new form of a battlefield modelling and simulation and might be applied for several purposes: from a decision support, via a simulation knowledge acquisition to a simulation training. In the conducted research, VBS2 has been enhanced with the additional behaviour algorithms that are typical for the situations resulting from new types of dangers for soldiers and civilians: anti-war demonstration, stroll, stroll with random selection of the next step and hiding in a building. These behaviour schemas are associated with another two algorithms of reaction to potentially dangerous incidents. One of them is related to units which are hidden in a building, while the other refers to units which operate outside. Both algorithms are designed separately for each the unit from the group.

The next part concentrates on merging two standalone simulation environments using the publishing/subscribing concept from the HLA standard (IEEE 1516). Some information for the simulation (e.g. units' structure, distribution or tasks) is produced by SSWSO Zlocien and published via the HLA RTI to VBS2 system, so both the systems should be HLA federates. SSWSO Zlocien has been built according to HLA rules. However, VBS2 had to be adapted, therefore we propose a proxy software called a *proxyderat* (a proxy for a federate). In contrast to the gateway called *LVC Game* (supplied by the manufacturer of the VBS2) our approach has been created in order to cooperate with constructive and event-driven simulations, and it does not exclude the possibilities of interactions with real-time simulations. Furthermore, a very important advantage is an ability to configure an exchanged data model and to manage a logical simulation time.

2 Modelling Civilian Activities

The new simulation models of soldier and civilian behaviours, regarding the asymmetric urban operations, enhance the VBS2 capabilities and improve, or even enable, analysis of asymmetric military urban threats. A basic way of creating new models is to use Mission Editor where actions (e.g. movement) of groups and individual units can be designed with so-called waypoints and triggers. A waypoint requires setting values of the following parameters: a group formation, a movement speed, a type of behaviour (e.g. careless or aware of danger) and rules of engagement (e.g. “fire at will”). The most important parameter is a type of waypoint which defines what kind of actions should be performed in that specific point. The following types of waypoints are available by default:

- *Move* – movement of a group or a unit to point placed at an appropriate distance;
- *Destroy* – destruction of an object pointed by a waypoint;
- *Get in* – board free space in a vehicle attached to a waypoint or in vehicles located in close proximity to waypoint’s location;
- *Get out* – disembark from vehicles in a location indicated by the waypoint;
- *Join, Join & Lead* – actions resulting in merging groups and choosing a leader;
- *Load, Unload* – actions to safely get in/get out of the particular vehicle;
- *Transport unload* – similar to “Unload” – only for units not belonging to crew;
- *Hold* – maintaining the point of particular grid references at any cost;
- *Guard* – protect locations from hostile units, connected with the “Guarded by”;
- *Sentry* – identification of an unknown unit in the given location and, depending on the results, attack or move to a next waypoint;
- *Support* – offer services to units which make a “Call support” request via radio;
- *Dismissed* – actions of units being off-duty (walking slowly or sitting down);
- *Cycle* – choosing the nearest waypoint other than previously selected.

Regardless of user-defined movement schemas, VBS2 provides default set of reactions to certain events. For example, for a civilian group there have been defined actions such as laying down on the ground (as a reaction to being suppressed by incoming fire) or running away (when a danger situation is over).

The behaviour models mentioned above are strongly oriented solely on military units. Additionally, the default reactions for events are not appropriate to creating the accurate behaviour of larger group or greater number of groups. Notwithstanding, the most convenient way of development there is a usage of script files based on a syntax of SQF scripting language. The other way to modelling different states and transitions is Finite-State Machine supported by the VBS Real Virtual Simulation Engine.

Finally, during the research we have conducted VBS2 has been enhanced with the new behaviour models: anti-war demonstration, stroll, stroll with random selection of the next step and hiding in a building.

2.1 Anti-war Demonstration

The purpose of this algorithm is to represent anti-war demonstration on areas affected by a military conflict. A demonstration is realized peacefully without using force

against other units participating in a scenario. An input data contains markers which define constraints of movement route, apart from group conducting a demonstration. The general idea is to move the whole group between markers and perform an appropriate set of animations during regular intervals. Commands are being executed in one loop as long as at least one unit in a group is alive. The algorithm steps are:

- Optionally – when a timer counts down to zero, each unit in the group randomly chooses one from the available animations. Any animation is performed using “switch-move” command (at the moment, if unit is playing a move it is immediately terminated and switched for a new animation).
- Direction of movement is being checked (whether group is moving towards “start” marker or “end” marker) and a destination marker is being determined.
- Command distance from a leader’s position to a marker is being calculated.
- Direction from a leader’s position to a destination marker is being calculated.
- Depending on a distance to a destination marker:
 - if the distance is shorter than a specified boundary value a position indicating next move’s destination is set to destination marker’s coordinates and direction is switched to the opposite one.

In the opposite case:

- position indicating next move’s destination is calculated using a function which returns coordinates of position relative to group leader, direction to destination marker and specified step distance.
- The whole group is being ordered to move to a calculated position.
- Timer is being updated.

2.2 Stroll with Random Selection of the Next Step

The basic algorithm represents a slowly moving group of people. Its actions are based on a repeated steps of planning and performing movement using a route defined by markers (supplied as an input data). Contrasting to the previous algorithms, a group moves in a crowd formation – it means that a group does not wait for the units that have not reached the given waypoint yet. The algorithm can be described as the loop:

- Marker pointed in an array (by index value) is being set as a destination marker.
- Distance from a leader’s position to a destination marker is being calculated.
- Direction from a leader’s position to a destination marker is being calculated.
- Depending on distance to a destination marker:
 - if that distance is shorter than a specified boundary value a position indicating next move’s destination is set to destination marker’s coordinates;

In the opposite case

- position indicating next move’s destination is calculated using a function which returns coordinates of position relative to group leader, direction to destination marker and specified step distance.
- The whole group is being ordered to move to a calculated position.

Adding a random selection of the next step makes each unit performs separately and independently these steps. It models the decision making process made by individuals independently of a group leader.

2.3 Reaction to Dangerous Incident

The algorithm of appropriate reaction for a detected event is executed concurrently with the models already described. It has an ability to control the unit and to suspend execution of movement. The basic factor a reaction's selection is affected is a stress level which is multiplied by a degree of vulnerability to panic. A stress level is increased when a potentially dangerous event is detected and gradually decreased when a danger is no more present.

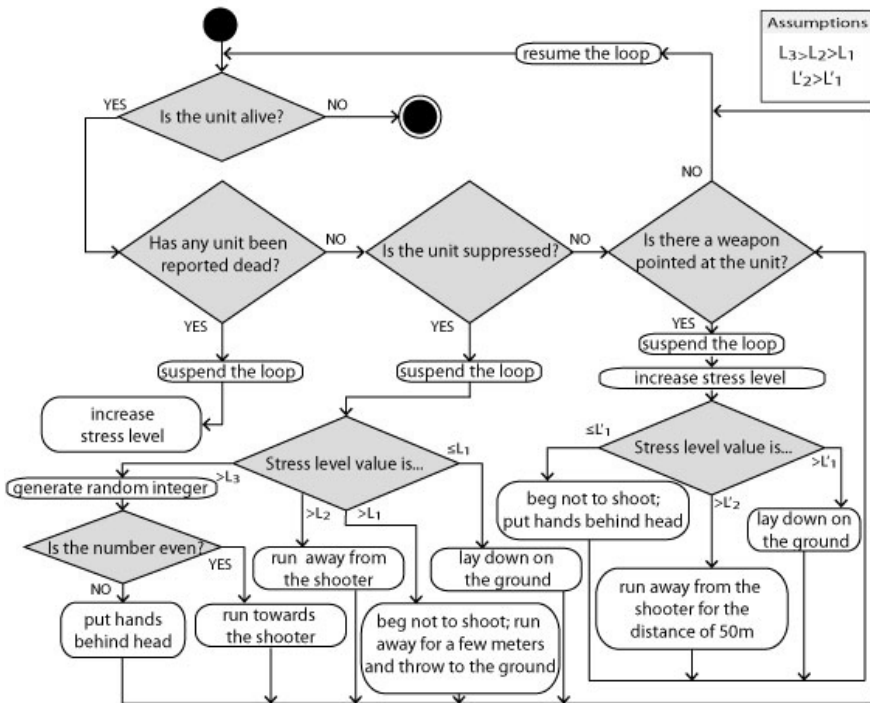


Fig. 1. The algorithm of reaction to potentially dangerous incidents

Situation where a unit is suppressed by an incoming fire is recognized by capturing an event “Suppressed”. One of actions performed at that moment is increasing stress level value. L_x and L'_x are boundary values that determine ranges of stress level values. Each range is connected with corresponding reactions on different stress levels. When a stress level value reaches its maximum, reactions are irrational and might increase a degree of danger. E.g., a person tries to run away from a danger (e.g. soldier firing a rifle) but it moves in a wrong direction, towards a source of a danger.

2.4 Hiding in Building

In the algorithm a group is to be placed inside a building “game logic” with a position set by finding “house” object which is the nearest to “game logic”. Units are being moved if there are enough positions to locate the entire group far from windows and outer walls in four-unit subgroups. It is a result of minimizing a risk of injuries or death. Decisions concerning reaction are being made based on a stress level and a condition of a building.

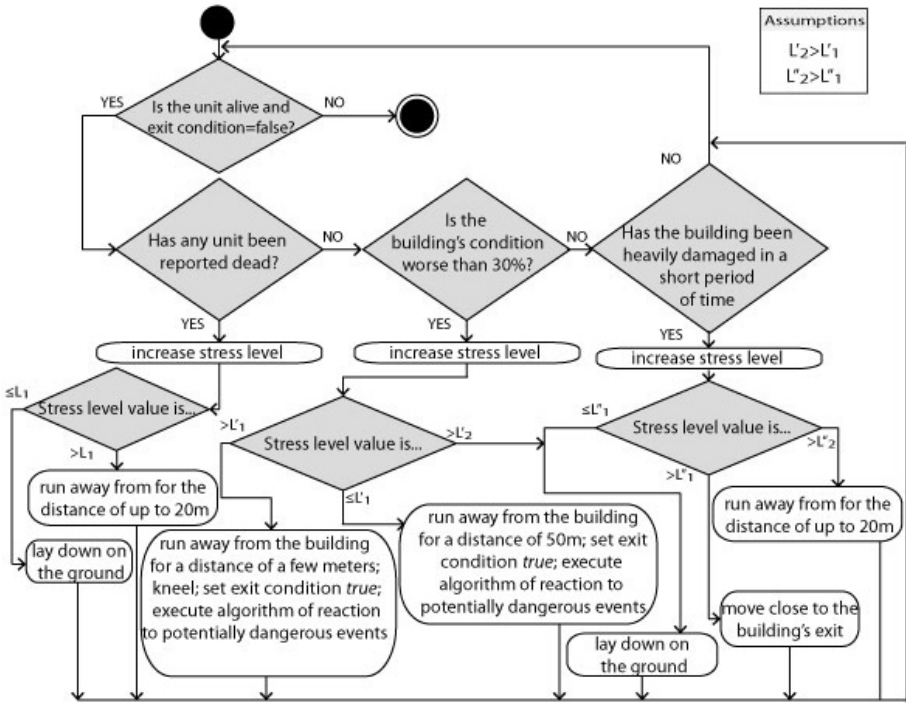


Fig. 2. The algorithm of hiding in a building and reaction

3 Integrated Simulation Environment

VBS2 system has in its offer a software gateway called LVC Game. It enables an integration with HLA or DIS standards. It passes the data in a real-time without a time stamp, hence its main application is to be used exclusively with real-time virtual simulators. Furthermore, it is limited to a predefined model of data being exchanged within the federation. The proposed approach has significant advantages: ability to configure exchanged data model and ability to manage a simulation time.

Physically the developed proxyderate is a component which is attached to the VBS2 runtime process when it starts (as a plug-in). It consists of two components:

PluginManager and VBSFederate. The first one is a VBS2 proxy and the second is a typical HLA federate. Simplified architecture is shown at Fig. 3.

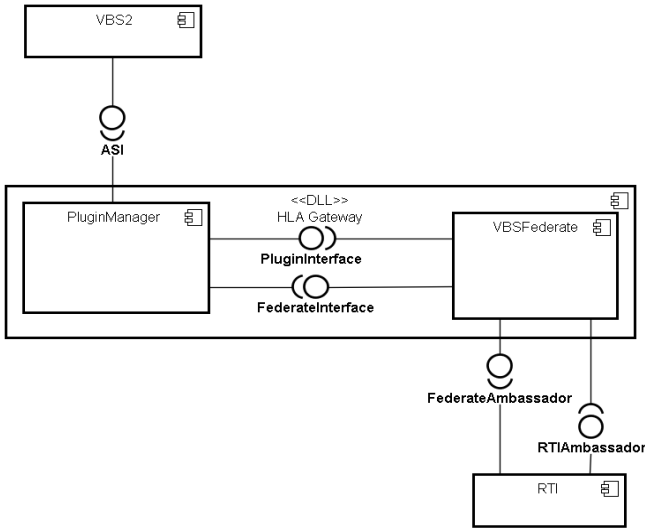


Fig. 3. The high-level architecture of the proxyderate

PluginManager is responsible for receiving simulation data from VBS2 and making necessary changes to publishing them into RTI. For these purposes, it uses the ASI Interface (Application Scripting Interface) which allows any external programs to execute scripting operations. VBSFederate is responsible for communication with the RTI via the two directed interfaces: RTIAmbassador and FederateAmbassador. The both components communicate with each other using FederateInterface and PluginInterface (may also be used by components other than VBSFederate).

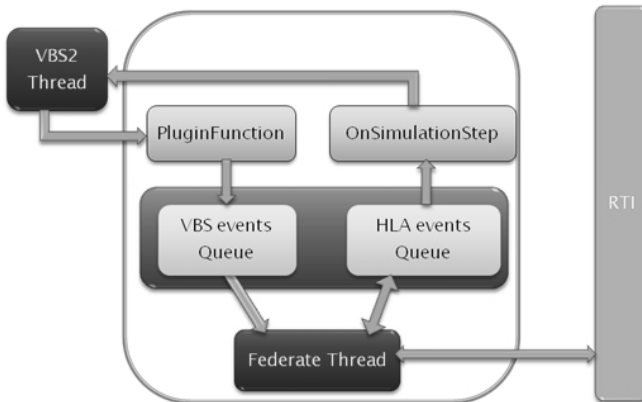


Fig. 4. Schema of data exchange between RTI and VBS2

A schema of exchanging data between VBS2 and RTI is shortly presented on Fig.4. The event queues (VBS events Queue and HLA events Queue) create a cache for simulation events coming from both VBS2 and the federate threads. VBS2 creates an event and calls PluginFunction() in order to classify a type of event and add to VBS events Queue. The external events, created by the rest of federation, are sent by RTI to the HLA events Queue and subsequently identified, classified and populated. All events are associated with timestamps (time of occurrence), thus VBSFederate component can receive and publish them in a proper order and at the correct simulation time. These activities are crucial to cooperation between VBS2 and event-driven or constructive simulations. On the other hand, this solution allows cooperation with real-time simulations since VBSFederate component is able to publish events immediately after receiving.

Comparing the proxyderate with VBS LVC Game, a possibility to work with different HLA Federation Object Models should be emphasized. The presented solution provides a mechanism to mapping attributes of HLA FOM objects to the attributes of VBS2 objects in a case when there are two different models of the same real objects. Proxyderate is responsible for maintaining the attribute’s states in both models.

An example at Fig.5 illustrates the mapping concept. In detail, for each VBS unit an eventhandler is attached at a creation time: Killed, HitPart and Damaged. An event occurrence causes a specified reaction, e.g., if the event Killed occurs the proxyderate will send an interaction to the rest of federation. Similarly, if Damaged or HitPart occurs HLA FOM object attributes will be updated.

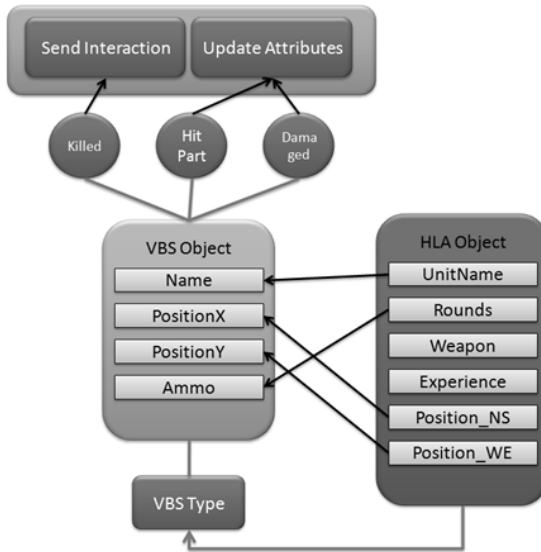


Fig. 5. The mapping concept of HLA FOM and VBS objects

5 Numerical Examples

The presented concept establishes a simulation environment with a federated structure that allows the interchange of simulation data using the proxyderate and HLA protocol, and assuming negligible delays. This assumption is particularly significant in a real-time simulation. In order to verify efficiency of the solution we developed, two different simulation scenarios have been built. The first one describes a direct combat engaged two 10-man groups, since the second one reflects a two 100-man groups. Both scenarios have been played in two experiment configurations: with and without proxyderate. A hardware configuration was typical: IntelCore2 2,53GHz, 2 CPU cores, RAM 4096Mb and 1066MHz,. The main measure is a delay between the two simulation steps.

Statistics gathered during the first scenario are as follows:

Table 1. Outcomes for the case: ``10 on 10'' fight

	Proxyderate - disabled	Proxyderate - enabled
Number of simulation steps	1000	1000
The duration of the experiment	39,2 s	39,4 s
The average time interval between simulation steps	0,0392 s	0,0394 s
	<i>Average delay</i>	<i>0,0002 s</i>

The first conclusion is that in case of both configurations the experiment completed almost at the same time. Delays resulted from simulation steps are almost imperceptible – average is 0.0002 s.

Table 2. Outcomes for the case: ``100 on 100'' fight

	Proxyderate - disabled	Proxyderate - enabled
Number of simulation steps	2000	2000
The duration of the experiment	110 s	118 s
The average time interval between simulation steps	0,055 s	0,059 s
	<i>Average delay</i>	<i>0,004 s</i>

In this case, the differences between both configurations are more noticeable: the absolute difference is 8 s, however the relative is only 7%. Although the average delay (0,004s) is higher than in the previous scenario, it is still acceptable as that is below the value of inertia of adjustment of the eye for clear seeing (0,01s).

6 Summary

The purpose of the studies has been to get better understanding and to provide adequate simulation of behaviour as observed in nature urban environment. As a result, VBS2 has been enhanced with the new behaviour models, dedicated for both military and civilian groups.

The presented proxyderate gives a possibility to use during a simulation different HLA Federation Object Models. It has been applied in a distributed simulation prototype designed especially for the army, however it could be also very useful for the security centres. This solution is flexible and adaptable to different models of data exchanged in a federation. Consequently, it makes possibilities to extend a simulation environments built on the basis of the HLA rules.

The distributed military simulation augmented by computational collective intelligence we proposed makes military training, operation planning and decision making more effective and therefore improves their quality. Notwithstanding, it is necessary to re-evaluate organization, methodologies and training scope to make sure that the required needs of urban operations are met.

Acknowledgments. This work was partially supported by the research projects: of the European Defence Agency (under the Call “Mission Planning/Training in an asymmetric environment” and “Secured Tactical Wireless Communications”) no. A-0937-RT-GC: Asymmetric Threat Environment Analysis – “ATHENA” and by project GD-651/2011/WCY of Cybernetic Faculty at Military University of Technology.

References

1. Antkiewicz, R., Kulas, W., Najgebauer, A., Pierzchała, D., Rulka, J., Tarapata, Z., Wantoch-Rekowski, R.: The Automation of Combat Decision Processes in the Simulation Based Operational Training Support System. In: Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA), Honolulu, Hawaii, April 1-5 (2007) ISBN 1-4244-0698-6
2. Pierzchała, D.: Designing and testing method of distributed interactive simulators. In: Proceedings of the 15th International Conference on Systems Science, Wroclaws (2004) ISBN 83-7085-805-8
3. Najgebauer, A., Antkiewicz, R., Pierzchała, D., Kulas, W., Rulka, J., Wantoch-Rekowski, R.: Modelling and simulation of C2 processes based on cases in the operational simulation system for CAXes 4(652),LVII (2008)
4. Kuhl, F., Weatherly, D., Dahman, J.: Creating Computer Simulation Systems - An Introduction To The High Level Architecture, PH PTR (1999) ISBN 0-13-022511-8
5. Schut, M.C.: Scientific Handbook for Simulation of Collective Intelligence, Version 2 (2007), <http://www.sci-sci.org/>

Time Based Modeling of Collaboration Social Networks

Gabriel Tutoky and Ján Paralič

Dept. of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice,
Letná 9, 040 01 Košice, Slovakia
{Gabriel.Tutoky, Jan.Paralič}@tuke.sk

Abstract. This article describes basic approaches for modeling of collaboration networks. We discuss typical ways of modeling collaboration networks and we also propose a new extension in weighting ties among event participants and the idea of aging of the ties among collaborators. Classical as well as proposed approaches to weighting of ties in collaboration networks are experimentally evaluated on real data set and compared with peoples' opinions expressed in a targeted inquiry.

Keywords: social network analysis, collaboration networks, networks modeling, network projection.

1 Introduction

In recent years, many “social networks” have been analyzed like various Internet communities, email networks, peer-to-peer networks, telephone call graphs or train routers [1]. All of these networks are interesting in some of their specific aspects and they provide a notable data source for network analysis. There are usually large-scale networks with thousands of nodes and edges. Analysis of these networks, usually based on global properties, can lead to interesting and helpful results. Nevertheless, there exist many different situations in network analysis where data used for analysis of these networks did not carry sufficient information, e.g. temporal information is often neglected in these analyses. In this paper we describe a new approach how to model and analyze one particular type of social networks, affiliation networks, making use of more strands of additional information, including the temporal one.

One of interesting types of social networks are affiliation networks. An affiliation network is a network of actors connected by common memberships in groups/events such as clubs, teams, organizations or other activities. Affiliation networks are special type of two-mode social networks [2] where one mode is a set of actors, and the second mode is a set of events which are affiliated to the actors. The tie between actors and events are created, if actor is member of a group/participates on particular event. Affiliation networks describe collections of actors rather than simply ties between pairs of actors. Based on such an affiliation network we are able to derive connections among members of one of the modes based on linkages established through the second mode [2].

Affiliation networks were studied in past, e.g. studying attendance of women in social events [3], movies and their actors [4] or co-authorship network of scientists and their papers [1]. Whereas in the first two examples, the authors used unweighted representations of the networks, in the last work, the author introduced interesting approach for building of collaboration network where he used weighted ties between authors of the same paper. The weight of the tie between collaborated authors of a single paper is derived from count of the paper collaborators, and final weight of two collaborated authors is a sum of weights over all papers where authors collaborated. This approach allows finding the “most connected” scientists in the whole collaboration network.

In our work we build collaboration network of teenagers (the necessary background details are described in next section 2) based on their participations on educative-pedagogic workshops for evaluating the “most important” persons in desired time stamp. In network building process we used two different approaches: 1) modified weighting of the ties between collaborators based on weighting used by Newman in [1, 5] and 2) our weighting described in details in section 2.2. Both of these weightings are next evaluated taking into account also the temporal aspects. We proposed time dependent weights decreasing over time (see section 2.3). We assume that weight of two collaborators is dependent on the number of events’ participants and on the time interval between events where these actors collaborated together. Weight decreases with increasing number of event’s participants and also with increasing length of the time interval between two common events.

2 DAK Collaboration Network

DAK¹ – community network is a collaboration network of members (usually teenagers) of a non-profit organization dealing with organizing educative-pedagogic workshops for young people [6]. Usually there are organized around 10 workshops annually with 150 – 700 participants on single workshop. The number of participants depends on workshop’s type and duration. All participants of a single workshop are partitioned into smaller groups, usually with 8 – 12 members. Each group member cooperates with other group members and so there are established new social relations between group members. Generally there are two types of group members – *group participants* and *leader(s) of a group*. Participants are spending most of the time inside the group (i.e. they usually do not establish social relations outside the group), but leader cooperates with other leaders and so they create another type of social relations. Additionally we recognize two types of groups – *participants’ group* and *organizers’ group*. In summary we have two types of groups and four types of group members: *base participant*, *base organizer*, *leader of participants’ group* and *leader of organizers’ group*.

Compositional attributes in DAK data set are available for both, actors and events. Actors are described by attributes such as date of birth (age) and gender; as well as by geographical attributes – city or area of living. Events are described by their type.

¹ <http://www.zksm.sk/>

We can recognize two main types of events – events for base participants and events for organizers. Events for organizers are next categorized by their types of activity like registration, security or accommodation event.

Moreover, temporal attributes are available together with compositional attributes, e.g. start and end of events/workshops. From these data we can derive several other attributes, such as “length of event” or “time of first visit” for particular actor. In our case it means the moment when the actor visited any event for the first time.

3 Collaboration Network Modeling

Collaboration network described above can be expressed for single workshop by a bipartite graph as representation of two-mode network. The first mode is a set of actors, and the second mode is a set of events which affiliate the actors. We represent each group on the single workshop as a single event, so for one workshop we obtain several events. Additionally we added two more events for representation of cooperation between leaders of participants and leaders of organizers.

One of the advantages of DAK data set is availability of temporal information in the data. We are able to track events in the time and recognize which events were organized in parallel and which sequentially. Also we are able to track participation of single actors on particular events.

2.1 Base Projection Onto One-Mode Network

Affiliation networks (two-mode representation) are most often projected onto one-mode networks where actors are linked to one another by their affiliation with events (*co-membership* or *co-operation*), and at the same time events are linked by the actors who are their members (*overlapping* events). This property is called *duality* of the affiliation network [2].

Usually, weights in both, affiliation (two-mode) and also in projected (one-mode) networks have binary values. The ties in the networks exist or not. In the step of projection of two-mode networks onto one-mode networks we can use different measures for weight definition, e.g. the ones summarized by Opsahl in [7]:

- Weight is determined as a count of participations (co-occurrences) – e.g. the count of events were two actors participated together, formalized expression is

$$w_{ij} = \sum_p 1, \quad (1)$$

where w_{ij} is the weight between actors (nodes of the first mode) i and j , and p are events (nodes of the second mode) where i and j participated together.

- Newman in [1, 5] proposed extended determination of weights while working with scientific collaboration networks. He supposes that strength of social bonds between collaborators is higher with lower count of collaborators on a paper and vice versa social bonds are lower with many collaborators on a paper. He proposed formula (see formula 2) for defining the weights among collaborators where N_p is the count of collaborators on paper (event) p .

$$w_{ij} = \sum_p \frac{1}{N_p - 1} \tag{2}$$

Till now we considered only binary two-mode networks and their projection to weighted one-mode networks. However, there exist also weighted two-mode networks, such as networks of online forums (weight is determined as count of posts or posted characters) or collaboration network described above and also in [8, 9]. So, both just presented measures for weight definition could be extended for weighted two-mode networks as follows:

- $$w_{ij} = \sum_p w_{j,p}, \tag{3}$$

where $w_{j,p}$ is the weight of j^{th} actor to p^{th} event where i and j participated together. This method differentiates how two particular actors interact with the common event, and projects it onto a directed weighted one-mode network. [7].

- In a similar way, the Newman’s method can be extended for projecting of two-mode networks. The weights are calculated by the following formula:

$$w_{ij} = \sum_p \frac{w_{j,p}}{N_p - 1}. \tag{4}$$

This formula would create a directed one-mode network in which the out-strength of a node is equal to the sum of the weights attached to the ties in the two-mode network that originated from that node [7].

2.2 Extension of One-Mode Projection

In the next two sections we describe our extensions of projection of two-mode collaboration networks onto one-mode networks. This step – projection of two-mode networks, has strong impact on analysis of collaboration networks. It is important step for creation of the most suitable network model by projection onto one-mode network.

At first, we propose new, more general weighting of the ties created among event participants as Newman’s weighting method. The reason is that Newman’s weighting method results in fast decreasing value with just a small increase of event participants (more than two). This can be good in some cases, but not in general for any collaboration network. We suggest using an exponential curve:

$$w_{ij} = \alpha^{2-N_p} \tag{5}$$

The weights are also here decreasing with increasing number of event participants. But parameter α can be adjusted with respect to particular type of collaboration network (and in such a way influence the shape of the exponential curve). Parameter α depends on collaboration type and it should be estimated by a domain expert e.g. with the following formula:

$$\alpha = \beta^{-2} \sqrt{\frac{1}{\gamma}}. \tag{6}$$

This formula enables easier set up of an optimal value of the parameter α for particular type of collaboration network because γ is the weight which should be established between β participants of an event in the collaboration network. For example in scientific collaboration network, the strength of collaboration ties among 8 scientists should be weaker (by Newman it is 0,14286), but e.g. in collaboration network of students or in the DAK network described above, the strength of the ties among 8 collaborators participating on the same event should be higher, e.g. 0,6². Number 2 used in the index of radical represents an “ideal” number of event participants, when the strongest ties are created among event participants (this is analogical to the Newman’s method).

2.3 Time Based Network Modeling

Various collaboration networks contain time series data –usually the time of the event is known. It is reasonable to assume that the weight of ties created between participants of a common event will decrease over time. So, we propose time dependent weights in our representation of one-mode projected affiliation network – a kind of aging of the ties. This should be considered as similar approach to the one presented in [10, 11] where authors considered aging of the nodes in the context of citation networks. They describe node’s age as influence to the probability of connecting current node to new nodes in the network.

Our proposed weight aging method is based on assumption that past collaborations among network members are less important than lately created collaborations. These past collaborations after passing sufficient long time have no more influence in the present and they are next removed from the network – old ties (without refreshing) among collaborators are than “forgotten” in such a way. From the social network analysis point of view our proposal of aging of the edges can lead to new opportunities in network analysis:

- *Tracking collaborations over the time* – i.e. tracking of collaboration strength with passing time among selected actors of the network. This should provide detailed information describing evolution of cooperation among desired actors.
- *Creation of network snapshots in given time* – it allows us to obtain actual network state in desired time and consequently to analyze e.g. strongest collaborations in the network. It can lead to different results of network analysis because we do not consider older collaborations so high like last created. In collaboration network we are able to “view” still actual and (by our confidence) important collaborations among network members.

We have investigated humans’ forgetting curve described by Ebbinghaus in [12] which has exponential progress for possibility of using it for aging of edges (i.e. for decreasing of their weights) with passing time. Forgetting curve can be described as

$$R = e^{-\frac{t}{s}}, \quad (7)$$

² We consider interval $\langle 0, 1 \rangle$ for weight values where 0 represents weakest and 1 represents strongest tie.

where R is memory retention, S is the relative strength of memory, and t is time. By [12, 13] the forgetting curve decreasing rate depended on repetition of memory – in collaboration network context it is repetition of collaborations among actors. If actors collaborate together frequently (the collaboration strength grows quickly in short time) the edge aging is slower than aging in case of just one (or very sporadic) occurrence of mutual collaboration. Also we have investigated similar works with aging of nodes where in [11] authors studied aging of the nodes in scientific collaboration networks and they derive exponential progress of network aging – similar to the forgetting curve (formula 7).

In network modeling process we propose the use an exponential curve for modeling of edges aging described by formula

$$w_{ij}(t + \Delta t) = \left(\frac{e-1}{w_{ij}(t)} + 1 \right)^{-\frac{\Delta t}{S}}, \quad (8)$$

where $w_{ij}(t + \Delta t)$ is the weight after Δt time left after the last collaboration in time t among actors i and j . Formula $\left(\frac{e-1}{w_{ij}(t)} + 1 \right)$ expresses rate of the edge weight decreasing in passing time and S is the relative strength of collaboration.

2.4 Network Modeling Process

We decompose the network modeling process into the following steps:

- *Creation of two-mode network* – from available real data we created affiliation network with expert support.
- *Projection of two-mode network* onto one-mode network using the following alternative weighting schemes:
 - Simple weighting – each collaboration on an event has value 1
 - Newman's weighting – collaboration strength is derived from number of event participants by formula $\frac{1}{N_p-1}$ (see equation 2).
 - Our proposed weighting – collaboration strength is derived from number of event participants by means of equation 5; for α we used value 1,04.
- *Solitary network modeling over the time:*
 - Simple summation over all collaborations – collaborations computed in step before are now summed – see equation 1 for simple weighting case and eq. 2 for Newman's weighting.
 - Aging of edges – simulation of network edges aging, we created 24 network snapshots, each one for the time of a workshop and we derived collaboration strength before and after the workshop. Collaboration strength between two selected actors is depicted on the following figure 1 for different types of analyzed weighting schemes.

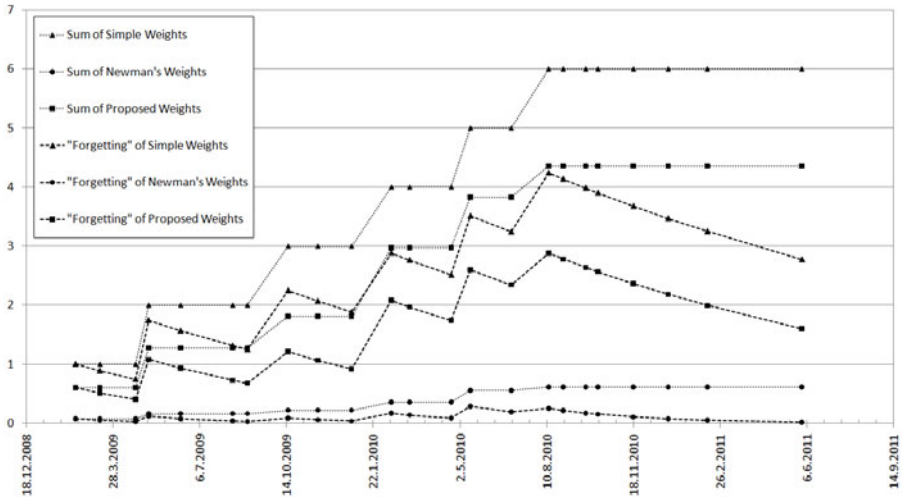


Fig. 1. Simple summing of collaboration weights (dotted lines) – collaboration weight between two selected actors never decreases; “Aging” of collaboration weights (dashed lines) – in the case of high frequency of collaboration (e.g. between dates 22.1.2010 and 10.8.2010) weight is increasing also relatively fast, but in case of no collaboration (after date 10.8.2010) weight is decreasing.

Triangle marked lines – weight is increasing in collaboration time with constant value 1; Circle marked lines – weight is increasing by Newman’s weighting, see formula (2) for dotted circle marked line; Square marked lines – weight is increasing with our proposed weighting, see formula (5) for dotted square marked line.

3 Evaluation

We implemented all methods for projection of two-mode networks onto one-mode networks presented above and we evaluated them for both variations of weighting of the ties – for simple summing of all collaboration weights; and also for aging of ties (collaborations) with passing time. In order to evaluate which of these approaches models best the reality, we used for comparison data gathered by means of targeted inquiry from 16 respondents who are actual members of the analyzed collaboration network. We selected such members of the network who know the network structure very well since a longer time period and follow activities of its members through organized workshops. Each of these respondents had to create a list with 30 most important persons in the network by his/her opinion. The goal was also to sort created list from most important to less important persons. As a result of this inquiry we obtained from all respondents altogether a list of 90 distinct actors which were mentioned 1 – 15 times by our respondents. For our evaluation we filtered this list for persons who were mentioned at least 4 times and next we ordered this list by average value of actor’s position assigned to him/her by respondents.

On the other side, for each particular model of collaboration network we obtained list of top 35, 30, 25 and 15 actors by Hubs and Authorities analysis [14, 15]. We

compared these lists with results from our inquiry. For each list size we evaluated the quality of estimation of most important actors by particular model of collaboration network (see figure 2) so that we first simply computed intersections between these lists (one gathered as a result from the inquiry process described above and the other one by calculation of Hubs and Authorities in case of particular collaboration network model) and expressed it in percentage of the whole. The results are graphically presented in figure 2.

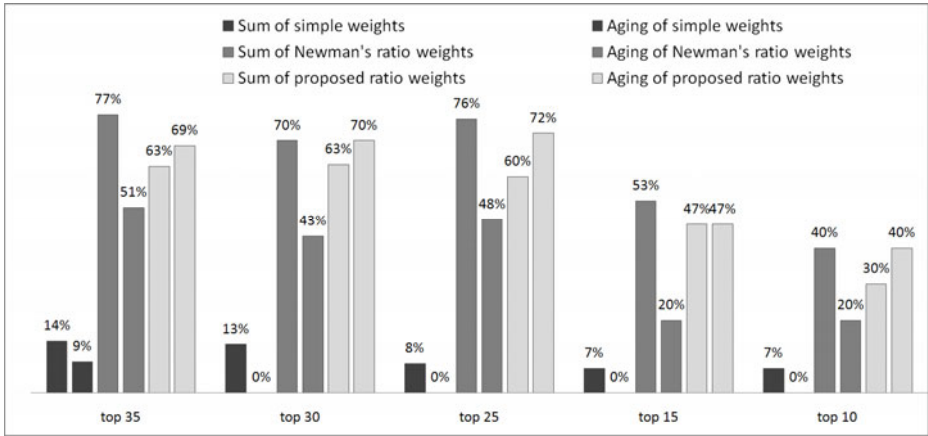


Fig. 2. Evaluation results of the most important actors in the network. Three weighting types – simple, Newman’s and our proposed weighting are distinguishable by gray tone. Left columns of the same tone displaying result for simple summing of weights over the time; whereas rights columns displaying results with aging of ties over the time.

This experiment confirmed our assumption that projection of two-mode network based on weighting with constant value 1 (formula (1)) cannot provide sufficient model of collaboration network (see dark columns in figure 2). On the second hand, this experiment showed unexpectedly high precision of results for Newman’s weighting of collaboration ties for simple summing of weights over the time (see all left middle gray columns). In this case we expected better results for Newman’s weighting than constant weighting, but we also expected higher precision of our proposed weighting. Our expectation was validated in case of aging of the ties where it has better results than Newman’s weighting (see lightest gray columns). In case of aging of the ties, our proposed weighting has better results than Newman’s weighting, especially for identifying 15 most important actors.

In the next step we evaluated mean absolute deviation of ordering of important actors so that we counted all differences (in absolute value) between ordering obtained from inquiry and from network analysis (see figure 3). We can see that the best results for aging have been achieved with our approach to weighting, and for simple summing our approach also achieved good results.

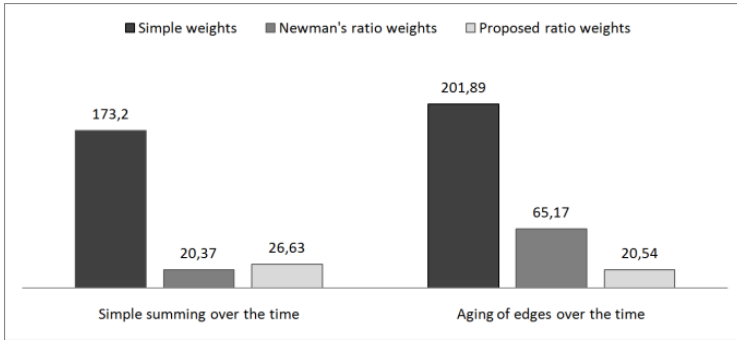


Fig. 3. Evaluation results of ordering of important actors by mean absolute deviation

4 Conclusion

In section 2.1 we described various methods for weighting of collaborations among event participants in collaboration networks and in section 2.2 we described one new method of weighting of the ties. In next sections 2.3 and 2.4 we described our method for modeling networks with passing time, where we proposed method for aging of the ties among collaborators. We next evaluated all presented methods on data from DAK collaboration network. Experiments brought positive results, showing that proposed type of weighting, especially in combination with aging resulted in very good results. But there is still space for further investigations of proposed methods.

In our future work we will evaluate different collaboration network models with further feedback from actors. Currently we are gathering data from respondents who are asked for expression of collaboration strength to persons from their collaborations. In addition they are also asked for expression of trend of collaboration strength in the last 2 years.

Acknowledgments. The work presented in this paper was supported by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic under grant No. 1/0042/10 (30%) and by the Slovak Research and Development Agency under the contract No. APVV-0208-10 (30%). This work is also the result of the project implementation Development of Centre of Information and Communication Technologies for Knowledge Systems (project number: 26220120030) supported by the Research & Development Operational Programme funded by the ERDF (40%).

References

1. Newman, M.E.J.: Who is the best connected scientist? A study of scientific coauthorship networks. *Complex Networks* (2004)
2. Wasserman, S., Faust, K.: *Social Network Analysis*. Cambridge University Press, Cambridge (1994)

3. Davis, A., Gardner, B.B., Gardner, M.R.: Deep South. A social Anthropological Study of Caste and Class. University of Chicago Press, Chicago (1941)
4. Watts, D.J., Strogats, S.H.: Collective dynamics of 'small-world' networks. *Nature* (1998)
5. Newman, M.E.J.: Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *The American Physical Society* 64 (2001)
6. DAK - Collaboration Network, data set of non-profit organization (2011), <http://www.domcek.org>
7. Opsahl, T.: Projecting two-mode networks onto weighted one-mode networks (2009)
8. Tutoky, G., Paralič, J.: Modelovanie a analýza malej komunitnej sociálnej siete. In: 5th Workshop on Intelligent and Knowledge Oriented Techniloties, Bratislava (2010)
9. Tutoky, G., Repka, M., Paralič, J.: Structural analysis of social groups in collaboration network. In: Faculty of Electrical Engineering and Informatics of the Technical University of Košice, Košice (2011)
10. Hajra, K.B., Sen, P.: Aging in citation networks. Elsevier Science, Amsterdam (2008)
11. Zhu, H., Wang, X., Zhu, J.-Y.: The effect of aging on network structure. *The American Physical Society* (2003)
12. Ebbinghaus, H.: Memory: A Contribution to Experimental Psychology. Teachers College, New York (1885)
13. Savara, S.: The Ebbinghaus Forgetting Curve – And How To Overcome It, <http://sidsavara.com/personal-productivity/the-ebbinghaus-curve-of-forgetting>
14. Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment. In: ACM-SIAM Symposium on Discrete Algorithms (1998)
15. Batagelj, V., Mrvar, A.: Pajek - Program for Analysis and Visualization of Large Networks, Ljubljana (2009)

Simulating Riot for Virtual Crowds with a Social Communication Model

Wei-Ming Chao and Tsai-Yen Li

Computer Science Department, National Chengchi University
64, Sec. 2, Zhi-Nan Rd. Taipei, Taiwan
{g9608, li}@cs.nccu.edu.tw

Abstract. In this work we attempt to build a communication model to simulate a large variety of collective behaviors in the course of riot formation for virtual crowd. The proposed crowd simulation system, IMCrowd, has been implemented with a multi-agent system in which each agent has a local perception and autonomous abilities to improvise their actions. The algorithms used in our communication model in IMCrowd are based heavily on sociology research. Therefore, the collective behaviors can emerge out of the social process such as emotion contagion and conformity effect among individual agents. We have conducted several riot experiments and have reported the details of the correlation between the severity of a riot and three predefined factors: the size of the crowd, relative size of the parties, and initial position distribution of the crowd. We have found that crowd density and party size symmetry do affect the number of victims at the end. However, the initial distribution of the two parties does not significantly influence the index (number of victims) at the end.

Keywords: Virtual Crowd, Collective Behavior, Social Communication, Agent-Based Simulation, Computer Animation.

1 Introduction

Many applications can be benefited from crowd simulation, including entertainment, urban planning, emergency evacuation, and crowd behavior research for social sciences. However most previous efforts in crowd simulation focused on generating plausible animations for applications targeting more on visual effects without considering how communication among the agents could affect the behaviors of a crowd. These models are in general not adequate for investigating complex crowd behaviors because psychological and social factors, such as perception, emotional status, and communication mechanism, are either rarely concerned or greatly simplified. In [3], we have developed a system, IMCrowd, to simulate collective behaviors of virtual crowds. IMCrowd is built with an agent-based modeling approach and allows user to customize suggestive messages and other parameters to yield different kinds of collective behaviors and scenarios.

In this paper, we conduct several riot experiments to study how the related factors affect the end result of casualty numbers. In the next section, we will review the

literature related to collective behaviors and crowd simulation. Section 3 and 4 describe the system architecture of IMCrowd and the experimental environment, respectively. Section 5 presents the result of the riot simulation while Sections 6 are the conclusions and future work.

2 Literature Review

Psychologists and sociologists have studied the behaviors of groups of people for many years. They have been mainly interested in the collective behaviors which emerge from individuals under the influence of others in unexpected, sudden and unusual situations, named mass or crowd. In this case, people who are in a crowd act differently towards people, compared to those who are alone. They seem to lose their individual identities and adopt the behavior of the crowd entity, shaped by the collective actions of others.

LeBon Gustave [9] claimed that there were three steps of consensus process for the “casual crowd” behavior. The first step is that a crowd is prone to accept suggestions from others via communication. The second step is emotional contagion, which means the emotion of an individual can be infected by other nearby people in the crowd. The third, after they are infected by each other, an individual may substitute his or her self-consciousness with the group consciousness at certain situation. That explains why some individuals do something irrational in a crowd but they never do that when they are alone. Although the propositions of LeBon were made from observations, emotion contagion has been verified to some degree by modern psychologists [1][5]. Furthermore, Blumer’s work argued that contagion occurred through “circular reaction” wherein individuals reciprocally respond to each other’s stimulation [2]. That is, when individuals are exposed to a contagion environment, they reflect others’ state of feeling and in so doing intensify this feeling as a form of positive feedback.

Granovetter proposed a threshold model [7] to explain why similar crowds may behave differently. He regarded that each rational individual’s decision about whether to act or not depends on what others are doing. The threshold in this model means the number or proportion of other persons who take the action before a given person does so. It depends on personality as well as surrounding situations. The perception of other agents’ behaviors via various ways of communications may have a “domino” or “bandwagon” effect on crowd behavior when the tipping point is reached (above the threshold value).

Referring to the work of Collin [4], we know that violence simulation is difficult but there are some interesting features that can determine whether violence will happen or not:

1) Violence is always in the form of a small proportion of people who are actively violent and a large number of the audience who behave nominally violent or emotionally involved such as making noise or just looking. 2) Emotional supporters provide Emotion Energy (EE) to the violent few for going into action against the enemy. In other words, the violent few could not be launched without the prior EE. 3) Moments of violence in a riot are scattered in time and space. The visual scene of a riot falls into four categories: *standoffs*, *attacks*, *retreats* and *victories*. Standoffs are

generally dense and unviolent yet. When actual fighting breaks out, in attacks and retreats, the scene always breaks up. And the victory part tends to reunite. 4) Bluster is often the first step in a fight, but also an attempt to scare the enemy or avoid being dominated for averting violence. So, the confrontation is usually bluster and gesture but usually leads to little real harm. 5) In a riot, people always act with a combination of rational calculation and socially based emotion. In addition, fighting is always in a form of attacking the weak. 6) Violence is the most dependent situational contingencies – the solidarity of one side suddenly breaking up into little pockets so that a superior number of the other side can isolate and beat up an individual or two separated from the opposite group.

Jager et al. [8] modeled clustering and fighting in two-party crowds. A crowd is modeled by multi-agent simulation using cellular automata with three simple rules: a restricted view rule, an approach-avoidance rule, a mood rule. A restricted view rule means that an agent can only monitor nearby agents. The approach-avoidance rule governs whether an agent moves toward the agents of other party or of its own party. The mood rule indicates that the approach-avoidance tendencies are susceptible to an aggression motivation. Simulation consists of two groups of agents of three different kinds: *hardcore*, *hangers-on* and *bystanders*. These different types of agents scan their environment with a different frequency, and this causes them to cluster and increases their aggression motivation in different speeds. The results show that fights typically happen in asymmetrical large groups with relatively large proportions of hardcore members.

3 Introduction to IMCrowd

As shown in Fig. 1, IMCrowd is a multi-agent system consisting of two main components -- Agent behavior model and Communication model. The agent behavior model implemented Reynolds's Steering force model [12] and flocking model [11] as well as the pedestrian behavior model [13] such that they can improvise their individual and group behaviors such as goal seeking, fleeing, obstacle avoidance, collision avoidance, and group movement in the continuous space. While the agent behavior model enables the agents to move autonomously, the communication model enable them to make social interaction with others and decide what action to take. The communication model is inspired by Quorum Sensing [10], which is a process that allows bacteria to communicate with others for collectively regulating their gene expression, and therefore their collective behaviors. With the communication model, agents can propagate their emotion, react to the conformity pressure, and take a proper action according to their surrounding situations.

In IMCrowd, there are two kinds of agents: *regular* agent and *special* agent. The regular agents belong to a group that moves together by following a designated leader. A regular agent may act with either individual mind or group mind. When an agent acts with an individual mind, it pursues specified goals in sequence while avoid collisions with other agents. When an agent loses its individual mind and assumes a group mind, it acts according to the collective behaviors assumed by other agents in the scene. In the current implementation of IMCrowd, three collective behaviors have been designed: *gathering*, *panic*, and *riot*. These behaviors are initiated by special

agents carrying the group emotion and attempting to infect other agents they encounter. In this paper, we focus on the experiments of simulating riots, which include complex collective behaviors such as *assembling*, *bluster*, *vandalism*, *assault*, and *flight*. We use the observation of Collins [4] to design a decision tree for action selection under the group mind in a riot as shown in Fig. 2.

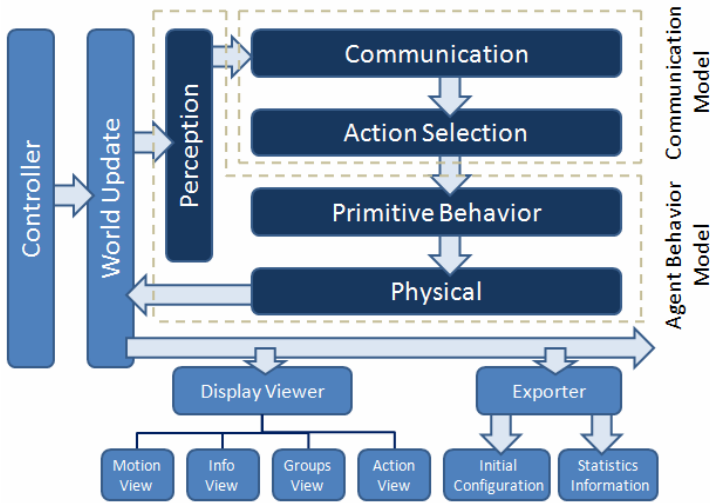


Fig. 1. System architecture of IMCrowd

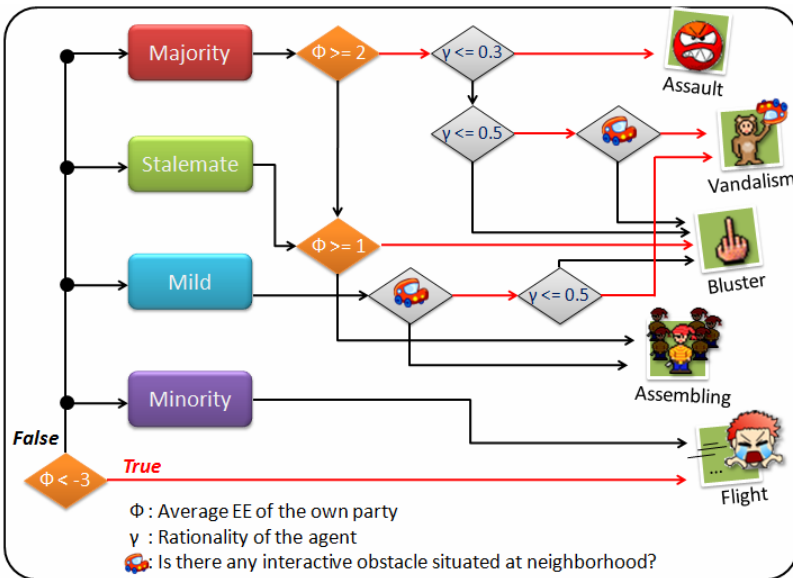


Fig. 2. Decision tree for action selection in a riot

The simulation environment for virtual crowd is in a continuous space that wraps over at the boundaries. The space is also overlaid with a 40 by 40 grid that is used to compute statistic information such as density, entropy, and superiority. IMCrowd provides an interactive interface for the placement of targets, obstacles and agents as shown in Fig. 3. There are two types of obstacles: *movable* and *unmovable*. Movable obstacles, such as cars, garbage cans, are objects that can become the target of vandalism while unmovable obstacles are environmental objects such as trees and fountains. The movable objects are used to simulate the window-breaking effect that causes the agents to reduce their rationality when they see objects being destroyed.

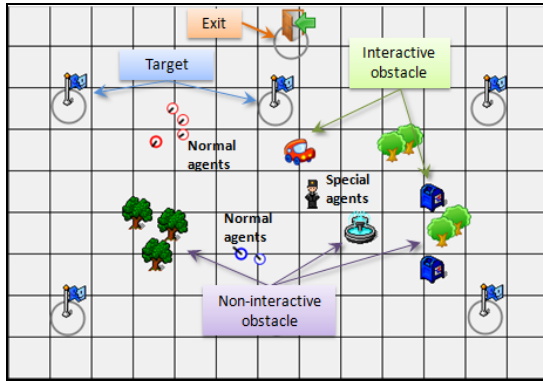


Fig. 3. Sample environment for crowd simulation in IMCrowd

4 Experiments of Riot Simulation

We have designed several experiments to study how various crowd factors affect the simulation results of a two-party crowd. These factors include *crowd size*, *symmetry of relative sizes*, and *initial distributions*. Two typical values are chosen for each factor. For example, two crowd sizes (*small*:100 and *large*:200) are used in the experiment, and the relative sizes are set with two values: *symmetric* (1:1) and *asymmetric* (3:1). Two types of initial distributions are also used: *well mixed* and *clustering*. Eight cases with variation of these three factors (2x2x2) were designed and labeled as shown in Table 1. For each case, we have designed ten scenes and run the simulation for 25,000 frames to collect experimental data. A major index on the behavior of the crowd in a riot is the number of the victims caused during the riot. The mean of this index out of the ten runs is computed for each case.

Table 1. Design of eight experimental cases

Crowd Size \ Position Distribution	100		200	
	Symmetrical (50 v.s. 50)	Asymmetrical (75 v.s. 25)	Symmetrical (100 v.s. 100)	Asymmetrical (150 v.s. 50)
Well-Mixed	A	C	E	G
Clustering	B	D	F	H

In Table 2, we show the comparison of the means on the number of victims under the eight cases. For the effect of crowd sizes, we found that large crowds always cause more victims than small crowds. This is due to the fact that large crowd means higher density in a limited space and higher density increases the speed of emotion cognition and the conformity effect in IMCrowd. It could also trap the agents into a positive-feedback loop that sustains their collective actions of riot and results in more casualties.

Table 2. Simulation results (number of victims) of the eight cases

		The Size of Crowd	
		100	200
Symmetrical	Well-Mixed	A 11.5	E 55.9
	Clustering	B 11.8	F 57.2
Asymmetrical	Well-Mixed	C 7.8	G 21.0
	Clustering	D 8.6	H 25.1

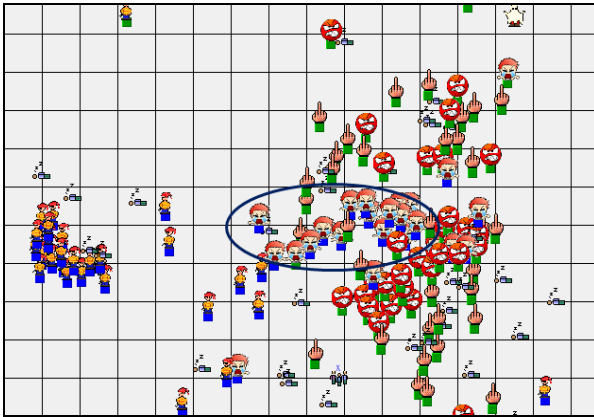


Fig. 4. Snapshots showing that the blue party is gradually surrounded by the opposite party after some confrontation

We also compare the symmetry on the sizes of the two parties. We found that when the size of crowd in each part is about the same (symmetric), there are more casualties than the asymmetric crowd, which is the opposite of the result reported in the literature [8]. From the simulation process, we observed that when the sizes of the two parties are about the same, it is more likely for them to gain local superiority in turn during the simulation. Due to high emotion energy and low rationality in such a situation, the agents are more likely to misjudge the global situation and select aggressive actions such as vandalism and assault and cause more casualties as a result. In Fig. 4, we show a situation where the agents in the blue party in the middle of the environment turn into relative inferiority in numbers, and they are gradually encircled and suppressed by the opposite party. Consequently, some of those agents who are locally inferior in numbers become victims. In contrast, if the size is

asymmetric, the agents in the inferior party are more likely to select the flight action and avoid confrontation with the superior party. A snapshot of the simulation showing the inferior green party taking a flight action is shown in Fig. 5.

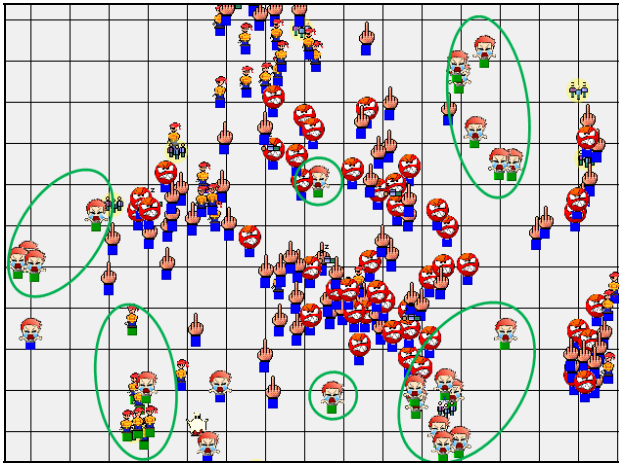


Fig. 5. Snapshot showing that the green party is taking a flight action because they are inferior in number

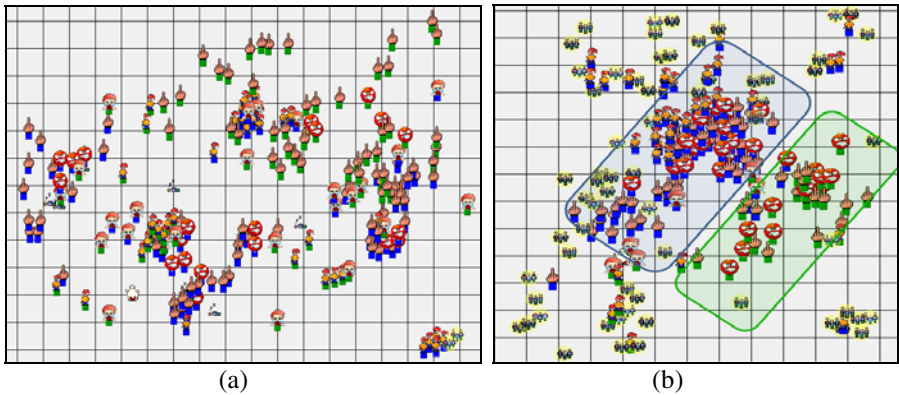


Fig. 6. Snapshot showing that (a) both parties are initially well-mixed and scattered in the environment and (b) both parties are initially separated into two clusters and the agents at the confrontation are more likely to assault the agents in the other party

We also have investigated the condition of different initial position distributions of the crowd: *well-mixed* or *clustering*. The well-mixed condition was produced by randomly assigning the position of each agent for both parties while the clustering condition was created manually to arrange the agents in a given region for each party. The experimental results show that the initial distribution has limited effects on the subsequent simulation as well as the resulting number of victims. By observing the

simulation process, we found that this is due to the fact that the agents have high degrees of mobility under the individual mind and there is enough time for the agents in each party to form groups before transforming to the group mind. Nevertheless, if the agents are initially cluttered, the number of victims is slightly higher than the situation of well-mixed. The situation is more significant when the density of the crowd is higher (200 agents) due to the fact that the mobility of the agents becomes less. A snapshot of the simulation in Fig. 6(a) shows that both parties are initially well-mixed and scattered in the environment. After some time, both parties are trapped into the melee and not able to congregate separately. On the other hand, the snapshot in Fig. 6(b) shows the situation where both parties are initially separated into two clusters and the agents at the confrontation are more likely to assault the agents in the other party.

5 Conclusion and Future Work

In this paper, we have presented a virtual crowd simulation system, called IMCrowd, that can simulate collective behaviors for virtual crowd with a communication model. We are able to simulate realistic crowd because we have enabled the agents with abilities at various levels such as spatial perception, autonomous movement, collision avoidance, emotion cognition, and group conformity. Based on the work of Collins on riot study [4], we have designed a decision tree model for the selection of actions for collective behaviors in a riot situation. We have designed eight cases in our experiments to study how three factors (size, symmetry, and initial distribution) of a crowd affect the severity of a riot. An interesting observation on the experiments that is different from the report by Jager [8] in the literature is that asymmetric/unbalanced group sizes do not necessarily result in more casualties if the agents in the inferior party are modeled with the ability to take the flight action and avoid confrontation.

Social scientists have developed several different theories for explaining crowd psychology. However, the simulation in IMCrowd is mainly based on the contagion theory and social imitation process for presenting the spontaneity of a casual and short-lived crowd without considering the cultural expectations, social background, and the motives and beliefs of participants. In addition, the individuals in IMCrowd only act with the rules we predefined and do not have the ability to learn new rules or establish new norms that emerges as the situation unfolds. Therefore, IMCrowd is not capable of simulating the social movements, racial hatred or the hostile that has been simmering for some time among groups of people. Instead, IMCrowd primarily focuses on the motion information or patterns in crowd dynamics such as panic, gathering and football hooligan riot.

The communication model that we have been presented in this work can be considered as the first model being used to reveal the emotion contagion and bandwagon effect in the dynamics of crowd simulation. Hence, many elaborations are possible. One of these possibilities is to differentiate the communication ability of the agent. As mentioned in the literature review, Gladwell [6] claimed that a few critical people who have distinctive personalities play the decisive role in inciting the social or behavioral epidemics. These key people usually have remarkable social skills and social contacts to effectively disseminate their influence. Collins [4] also mentioned

that most of people are not good at violence and what they manage to do depend on the emotion energy provided by other people. Nonetheless, a small portion of people is competently violence and has talent to whip up the emotion in the crowd for dominating their enemy. And this small group of people usually can disproportionately make a riot out of control. Therefore, we could try to equip the agents with different communication skills such that not all agents contribute the same influence to study the process of emotion contagion and bandwagon effect.

Acknowledgments. This research was funded in part by the National Science Council (NSC) of Taiwan under contract NSC 98-2221-E-004-008 and by the Top University Project of National Chengchi University, Taiwan.

References

1. Anderson, C., Keltner, D., John, O.P.: Emotional Convergence Between People over Time. *J. of Personality and Social Psychology* 84(5), 1054–1068 (2003)
2. Blumer, H.: *Symbolic Interactionism. Perspective and Method.* Prentice Hall, Englewood Cliffs (1969)
3. Chao, W.M., Li, T.Y.: Simulation of Social Behaviors in Virtual Crowd. In: *Computer Animation and Social Agents* (2010)
4. Collins, R.: *Violence: A Micro-Sociological Theory.* Princeton University Press, Princeton (2008)
5. Gaad, C., Minderaa, R.B., Keysers, C.: Facial expression: What the mirror neuron system can and cannot tell us. *Social Neuroscience* 2(3-4), 179–222 (2007)
6. Gladwell, M.: *The Tipping Point: How Little Things Can Make a Big Difference.* Little, Brown and Company, London (2000)
7. Granovetter, M.: Threshold Models of Collective Behavior. *The American Journal of Sociology* 83(6.I), 1420–1443 (1978)
8. Jager, W., Popping, R., van de Sande, H.: Clustering and Fighting in Two-party Crowds: Simulating the Approach-avoidance Conflict. *J. of Artificial Societies and Social Simulation* 4(3) (2001)
9. LeBon, G.: *The Crowd: A Study of the Popular Mind* (1895)
10. Miller, M.B., Bassler, B.L.: Quorum sensing in bacteria. *Annu. Rev. Microbiol.* 55, 165–199 (2001)
11. Reynolds, C.W.: Flocks, Herds and Schools: A distributed behavioral model. In: *The 14th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 25–34. ACM Press, New York (1987)
12. Reynolds, C.W.: Steering behaviors for autonomous characters. In: *Game Developers Conf.*, pp. 763–782 (1999)
13. Rymill, S.J., Dodgson, N.A.: A Psychologically-Based Simulation of Human Behavior. In: *Theory and Practice of Computer Graphics*, pp. 35–42 (2005)

Building Detection and 3D Reconstruction from Two-View of Monocular Camera

My-Ha Le and Kang-Hyun Jo

Graduated School of Electrical Engineering, University of Ulsan, Ulsan, Korea
lemyha@islab.ulsan.ac.kr, acejo@ulsan.ac.kr

Abstract. This paper proposes a method for building detection and 3D reconstruction from two-view by using monocular system. According to this method, building faces are detected by using color, straight line, edge and vanishing point. In the next step, invariant features are extracted and matching to find fundamental matrix. Three-dimension reconstruction of building is implemented based on camera matrixes which are computed from fundamental matrix and camera calibration parameters (essential matrix). The true dimension of building will be obtained if assume the baseline of monocular system is known. The simulation results will demonstrate the effectiveness of this method.

Keywords: Building faces detection, feature extraction and matching, two-view geometry, linear triangulation, 3D reconstruction.

1 Introduction

Three-dimensional objects reconstruction and distance/dimension measurement is one of important process in application of virtual environment, scene planning, and navigation of autonomous mobile robot. Some progress has been made in the reconstruction of 3D obtained during the last few years but they needed a large amount of work done by hand or apparatus, such as laser radar, and airborne light detection and ranging.

Three-dimensional reconstruction from two un-calibrated views has been deeply studied [1], [2], [3], [4], [5], [6], [7], [8]. For more clearly understand of this problem, it is possible to distinguish some proposed approaches into groups. One group of methods [9], [10], [11], [12], [13], [14] is based on scene knowledge or structure like parallelism, orthogonality and scenes with planes in order to perform camera calibration. However, not all scenes have this knowledge and the extraction of it could be lacking. Other group of methods based on self-calibration methods performs camera calibration by means of Kruppa's equations derived from the fundamental matrix [15], [16], [17], [18], [19] and approaches extracted from these equations [4], [5], [6], [7], [8], [20], [21], [22], [23]. With these methods, we do not need previous knowledge about the scene, but they have to employ epipolar geometry and estimate the fundamental matrix.

Without using any addition device, e.g. laser sensor out of single camera, our proposed method combine two approaches mentioned above. We utilize scene structure

to find object region in the images based on parallelism of line segment of object, also camera projection matrix will be derived from calibrated camera information and fundamental matrix. The flow chart of proposed method can be seen in Fig. 1. From monocular system, sequence image are obtain. Building faces detection methods were performed in our previous researches [24]. SIFT algorithm [25], [26] is applied to find invariant feature and matching problem. The estimation of fundamental matrix is performed base on 8-points algorithm [27]. Essential matrix and camera projection matrix are derived from computed fundamental matrix and camera calibration information. Finally, linear triangulation is last step to build 3D point of objects. The note in this final step is: the true information of objects can be obtained if we know the displacement of camera at adjacent moments, i.e., baseline. Our proposed method concentrate on building object and utilization of building face detection, i.e., corner points of building faces are extracted automatically, the 3D reconstruction with ambiguity up to scale and true information of building i.e. distance from camera to building, and dimension of building will be derived.

This paper is organized into 6 sections. The next section is summarization building face detection method. Section 3 is camera geometry and camera model, we also explain triangulation and true dimension measurement in this section. Experiments are showed in section 4. Paper is finished with conclusions in section 5.

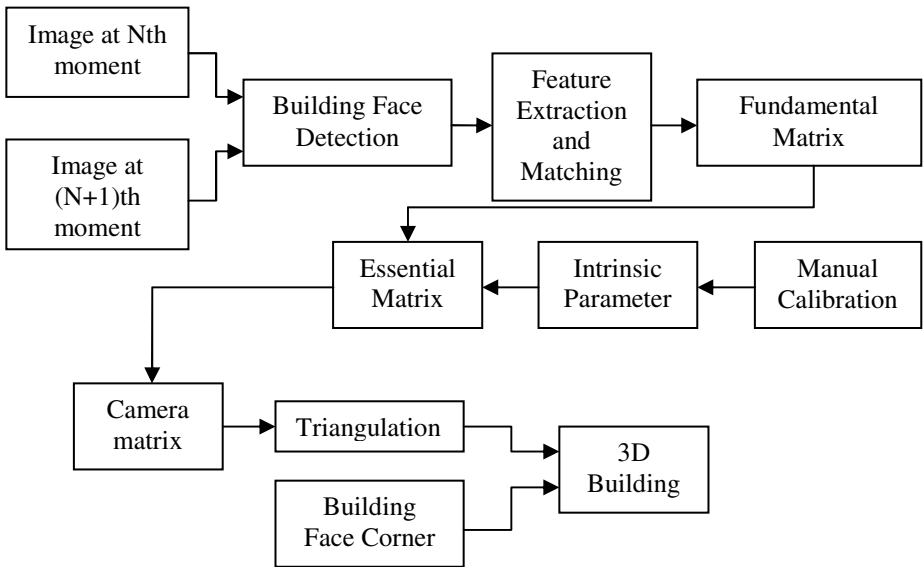


Fig. 1. Flow chart of proposed method

2 Building Detection Summarization

We use line segments and belongings in the appearance of building as geometrical and physical properties respectively. The geometrical properties are represented as

principal component parts (PCPs) as a set of door, window, wall and so on. As the physical properties, color, intensity, contrast and texture of regions are used. Analysis process is started by detecting straight line segments. We use MSAC to group such parallel line segments which have a common vanishing point. We calculate one dominant vanishing point for vertical direction and five dominant vanishing points in maximum for horizontal direction [28], [29]. A mesh of basic parallelograms is created by one of horizontal groups and vertical group. Each mesh represents one face of building. The PCPs are formed by merging neighborhood of basic parallelograms which have similar colors. The PCPs are classified into doors, windows and walls. Finally, the structure of building is described as a system of hierarchical features. The building is represented by number of faces. Each face is regarded by a color histogram vector. The color histogram vector just is computed by wall region of face. The overview of this method is show in flow chart of Fig. 2.

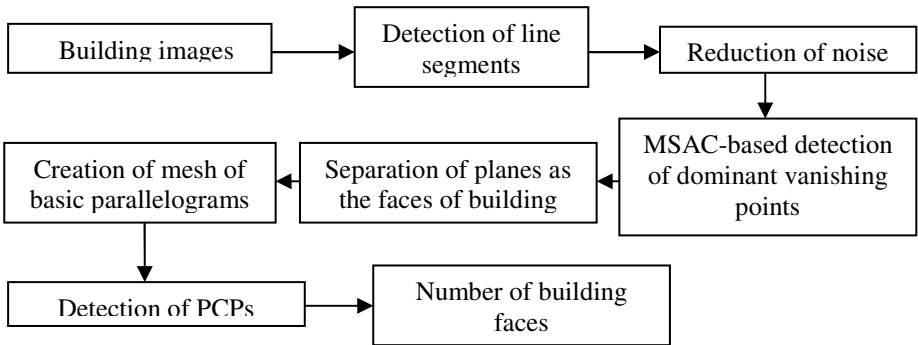


Fig. 2. Flow chart of building face detection

2.1 Line Segment Detection

The first step of the line segment detection is the edge detection of image. We used the edge detection function with Canny edge detector algorithm. The function is run in automatically chosen threshold. The second step is line segment detection following the definition: “A straight line segment is a part of edge including a set of pixels which have number of pixels larger than the given threshold (T_1) and all pixels are alignment. That means, if we draw a line through the ends, the distance from any pixel to this line is less than another given threshold (T_2)”.

2.2 Reducing the Low Contrast Lines

The low contrast lines usually come from the scene such as the electrical line, the branch of tree. Most of them usually do not locate on the edge of PCPs because the edge of PCPs distinguishes the image into two regions which have high contrast color. We based on the intensity of two regions beside the line to discard the low contrast lines.

2.3 MSAC-Based Detection of Dominant Vanishing Points

The line segments are coarsely separated into two groups. The vertical group contains line segments which create an actual angle 20° in maximum with the vertical axis. The remanent lines are treated as horizontal groups. For the fine separation stage, we used MSAC (m-estimator sample consensus) [27], [30] robustly to estimate the vanishing point.

2.4 Horizontal Vanishing Point Detection

Horizontal vanishing point detection is performed similarly to previous section. In reality, building is a prototypical structure where many faces and various color appear in images. Therefore, it is necessary to separate faces. We calculate five dominant vanishing points in maximum for horizontal direction.

2.5 Separation of the Planes as the Faces of Building

The vertical segments are extended by their middle points and vertical vanishing point. We based on the number of intersection of vertical lines and horizontal segments to detect and separate the planes as the faces of building. The results are showed in Fig. 3. The coarse stage of face separation is performed by the rule as following:

- a) If the same region contains two or more horizontal groups then the priority is given to a group with larger number of segment lines
- b) If two or more horizontal groups distribute along the vertical direction then the priority is given to a group with lower order of dominant vanishing point. The second stage is the recovery stage. Some horizontal segments which located on close the vanishing line of two groups are usually mis-grouped. Some segments instead of belonging to lower order groups, they are in higher order groups. So they must be recovered. The recovery stage is performed from the low to high. The third stage is finding boundaries of faces.

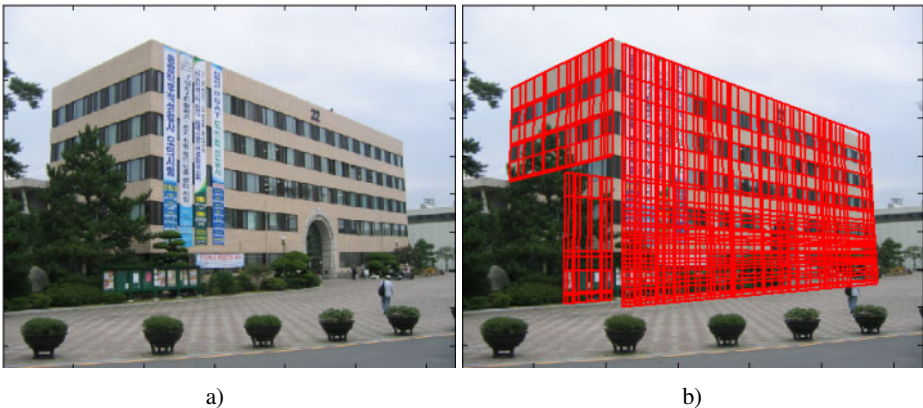


Fig. 3. Building detection result. a) Original image. b) Building face detection

3 Camera Model and Two-View Geometry

3.1 Camera Model

We use the projective geometry throughout this paper to describe the perspective projection of the 3D scene onto 2D images. This projection is described as follows:

$$x = PX \quad (1)$$

where P is a 3×4 projection matrix that describes the perspective projection process, $X = [X, Y, Z, 1]^T$ and $x = [x, y, 1]^T$ are vectors containing the homogeneous coordinates of the 3D world coordinate, respectively, 2D image coordinate.

When the ambiguity on the geometry is metric, (i.e., Euclidean up to an unknown scale factor), the camera projection matrices can be put in the following form:

$$P = K[R| -RT] \quad (2)$$

with t and R indicating the position and orientation of the camera and K , an upper diagonal 3×3 matrix containing the intrinsic camera parameters.

3.2 Feature Extraction and Matching

There are many kind of features are considered in recent research in feature extraction and matching include Harris [31], SIFT, PCA-SIFT, SURF [32], [33], etc. SIFT is first presented by David G Lowe in 1999 and it is completely presented in 2004. As we know on experiments of his proposed algorithm is very invariant and robust for feature matching with scaling, rotation, or affine transformation. According to those conclusions, we utilize SIFT feature points to find correspondent points of two-view images. The SIFT algorithm are described through these main steps: scale-space extrema detection, accurate keypoint localization, orientation assignment and keypoint descriptor. SIFT features and matching is applied for two view images as showed in Fig. 4. The result of correspondence point will be used to compute fundamental matrix described in the next step.

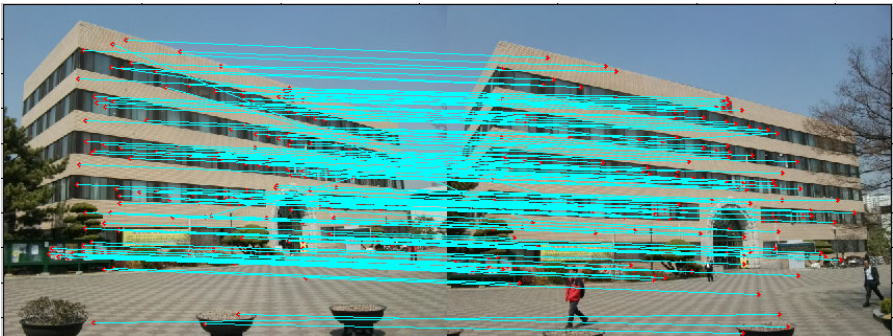


Fig. 4. SIFT feature extraction and matching

3.3 Two-View Geometry and Camera Matrix

The result of correspondence point in previous step will be used to compute fundamental matrix. The epipolar constraint represented by a 3x3 matrix is called the fundamental matrix, F . This method based on two-view geometry theory which was studied completely [27].

When the intrinsic parameters of the cameras are known, the epipolar constraint can be represented algebraically by a 3x3 matrix, called the essential matrix E . We have to do camera calibration to find this matrix. The good Matlab toolbox for doing camera calibration was provided by Jean-Yves Bouguet [34]. When we know camera intrinsic parameter, we can form the matrix K .

$$E = K'^T FK \quad (3)$$

Where E is essential matrix, K' and K are intrinsic parameters of frame 1 and 2. In the case of using the same camera, we have $K' = K$. The projection matrix of the first frame P is set follow this equation:

$$P = K[I|0] \quad (4)$$

The second projection matrix is found from four possible choices: $P' = (UWV^T|+u_3)$ or $P' = (UWV^T|-u_3)$ or $P' = (UW^T V^T|+u_3)$ or $P' = (UW^T V^T|-u_3)$, where U and V are found from SVD decomposition of E , u_3 is the last column of U and W . Only one of these four choices is possible for the second camera. We can find it by testing whether a reconstructed point lies in front of both cameras.

3.4 Linear Triangulation

Triangulation is the simplest but effective method to compute the 3D point X from the matching images points x and x' given two camera matrices. Difference with dense depth estimation or disparity map for image region, these linear triangulation are suitable for sparse points depth measurement. First, we have $x = PX$, but x is determined only up to scale in homogeneous coordinates. So we require that the vector x is collinear with vector PX by setting $x(PX) = 0$ which gives us two independent equations:

$$(P^{3T}X) - P^{1T}X = 0 \quad (5)$$

$$y(P^{3T}X) - P^{2T}X = 0 \quad (6)$$

where P^{iT} is the i th row of matrix P .

Similarly, we get another 2 equations from x' and P' and we establish an equation $AX = 0$. This equation is solved by SVD method to get X .

In order to reconstruct building face only and to show the results more clearly, we can refine our results by interpolation. We first need to pick an interest point in first frame and find its matching point in second frame automatically. Interest points are selected as corner points of building face which are detected in previous step. So we do the following steps:

- a) Pick up a corner point in first frame;
- b) Find out the epipolar line in second frame;
- c) Now we use a window sliding along the epipolar line, and find the sum of square error differences (SSD) compared to the one in first frame. The corresponding point is our matching point. SSD is given by:

$$D = \iint_R [I(R(x, y)) - I(R'(x', y'))]^2 w(x, y) dx dy \quad (7)$$

Where $I(R(x, y))$ is the intensity at the point (x, y) of a region R in one picture, and $w(x, y)$ is a Gaussian weighting function. A point (x', y') in R' that gives a minimum SSD from R is considered a feature match of (x, y) . The results of corner points matching from two-view are showed in Fig. 5.

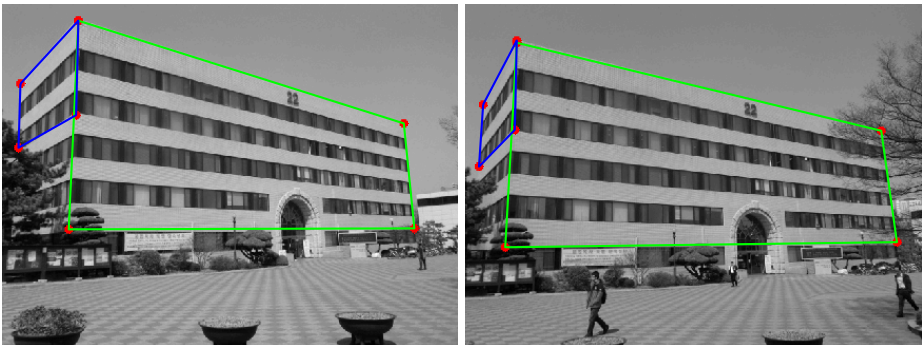


Fig. 5. Correspondence corner points of two-view

4 Experiments

We experimented on outdoor images which are acquired from CCD camera (Fujifilm, 3xf = 5.7-17.1mm 1:2.9-5.2). All result were simulated on Intel(R) Core(TM) i5 CPU 750@2.67 GHz with 3GB RAM under Matlab environment. The original images are color image with the size 640x480 pixel. The two faces of building are reconstructed by 8 points of two faces located on the corner.

For more clear visualization, we performed texture mapping process to map the building faces texture to reconstructed building face structure. On this process, each faces are divided into two Delaunay triangle plane because the results of 4 points triangulation didn't locate on the same plane. Then the texture on the building face regions is mapped to these triangles. Some distortions of face are also derived because of Delaunay triangle mapping in 4 points region as mentioned above. Fig. 6a, 6b, 6c and 6d showed difference angle of view of building faces which make good sense for determination the location of building faces in 3D space with texture mapping step.

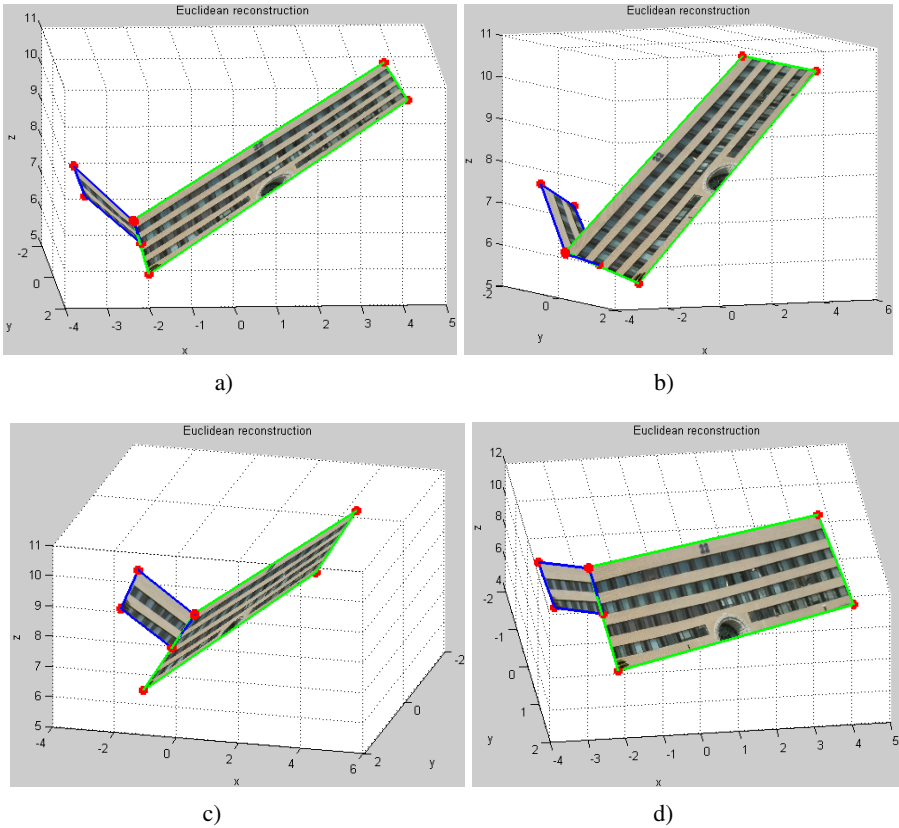


Fig. 6. Building face reconstruction results. a), b), c) and d) are several view angles of reconstructed faces

5 Conclusions

Three-dimension building reconstruction from two-view is presented on this paper. We implemented building faces detection to determine the exact corner points. The correspondence points of corner in the second view will be obtained from epipolar equation. Base on essential matrix and triangulation, the full information of building faces in 3D is extracted. With sparse corner points of building faces triangulation, this method can give the fast and correct information of object in 3D space. Also, the simulation results showed that: this method can overcome occlusion of building object in outdoor scene and robust with intensity condition because the process based on geometric information. Our future works focus on robust and stable estimation in camera self-calibration and faces reconstruction from line segment or plane in multi-views. We also will improve and develop this method for Omni-directional camera by using its video data in outdoor scene. Localization and mapping based on 3D information of surround environment will be considered for autonomous mobile robot application.

Acknowledgments. This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the Human Resources Development Program for Convergence Robot Specialists support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2010-C7000-1001-0007).

References

1. Longuet-Higgins, H.C.: A computer algorithm for reconstructing a scene from two projections. *Nature* 293, 133–135 (1981)
2. Hartley, R.I.: Estimation of relative camera positions for uncalibrated cameras. In: Sandini, G. (ed.) *ECCV 1992*. LNCS, vol. 588, pp. 579–587. Springer, Heidelberg (1992)
3. Hartley, R.I.: Euclidean reconstruction from uncalibrated views. In: *Proceedings of the Second Joint European - US Workshop on Applications of Invariance in Computer Vision*, pp. 237–256. Springer, London (1994)
4. Heyden, A., Astrom, K.: Euclidean reconstruction from constant intrinsic parameters. In: *ICPR 1996: Proceedings of the 1996 International Conference on Pattern Recognition (ICPR 1996)*, vol. I, p. 339. IEEE Computer Society, Washington, DC, USA (1996)
5. Bounoux, S.: From projective to euclidean space under any practical situation, a criticism of self-calibration. In: *ICCV 1998: Proceedings of the Sixth International Conference on Computer Vision*, p. 790. IEEE Computer Society, Washington, DC, USA (1998)
6. Sturm, P.: A case against kruppa's equations for camera self-calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* 22(10), 1199–1204 (2000)
7. Ma, Y., Koeka, J., Sastry, S.: Optimization criteria and geometric algorithms for motion and structure estimation. *Int. J. Comput. Vision* 44(3), 219–249 (2001)
8. Kanatani, K., Nakatsuji, A., Sugaya, Y.: Stabilizing the focal length computation for 3-D reconstruction from two uncalibrated views. *Int. J. Comput. Vision* 66(2), 109–122 (2006)
9. Caprile, B., Torre, V.: Using vanishing points for camera calibration. *International Journal of Computer Vision* 4, 127–140 (1990)
10. Triggs, B.: Autocalibration from planar scenes. In: Burkhardt, H.-J., Neumann, B. (eds.) *ECCV 1998*. LNCS, vol. 1406, pp. 89–105. Springer, Heidelberg (1998)
11. Cipolla, R., Drummond, T., Robertson, D.: Calibration from vanishing points in image of architectural scenes. In: *The 10th British Machine Vision Conference (1999)*
12. Zhang, Z.: A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(11), 1330–1334 (2000)
13. Svedberg, D., Carlsson, S.: Calibration, pose and novel views from single images of constrained scenes. *Pattern Recogn. Lett.* 21(13-14), 1125–1133 (2000)
14. Kösecká, J., Zhang, W.: Video compass. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2353, pp. 476–490. Springer, Heidelberg (2002)
15. Maybank, S.J., Faugeras, O.D.: A theory of self-calibration of a moving camera. *Int. J. Comput. Vision* 8(2), 123–151 (1992)
16. Luong, Q.-T., Faugeras, O.D.: The fundamental matrix: theory, algorithms, and stability analysis. *Int. J. Comput. Vision* 17(1), 43–75 (1996)
17. Zeller, C., Faugeras, O.D.: Camera self-calibration from video sequences: the kruppa equations revisited. Technical Report RR-2793, INRIA, France (1996)
18. Luong, Q.-T., Faugeras, O.D.: Self-calibration of a moving camera from point correspondences and fundamental matrices. *Int. J. Comput. Vision* 22(3), 261–289 (1997)

19. Ma, Y., Vidal, R., Kösecká, J., Sastry, S.: Kruppa equation revisited: Its renormalization and degeneracy. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 561–577. Springer, Heidelberg (2000)
20. Hartley, R.I., Silpa-Anan: Reconstruction from two views using approximate calibration. In: Proceedings 5th Asian Conf. Computer Vision, Melbourne, Australia, vol. 1, pp. 338–343 (2002)
21. Pollefeys, M., Koch, R., Gool, L.V.: Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *Int. J. Comput. Vision* 32(1), 7–25 (1999)
22. Sturm, P.: On focal length calibration from two views. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA, vol. II, pp. 145–150. IEEE Computer Society Press, Los Alamitos (2001)
23. Sturm, P., Cheng, Z., Chao, P.C., Neow Poo, A.: Focal length calibration from two views: method and analysis of singular cases. *Computer Vision and Image Understanding* 99(1), 58–95 (2005)
24. Trinh, H.H., Kim, D.N., Jo, K.H.: Supervised Training Database for Building Recognition by Using Cross Ratio Invariance and SVD-based Method. *International Journal of Applied Intelligence* 32(2), 216–230 (2010)
25. Lowe, D.: Object recognition from local scale-invariant features. In: Proc. of the International Conference on Computer Vision, pp. 1150–1157 (1999)
26. Lowe, D.: Distinctive Image Features from Scale-Invariant Interest Points. *International Journal of Computer Vision* 60, 91–110 (2004)
27. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2004) ISBN: 0521540518
28. Trinh, H.H., Jo, K.H.: Image-based Structural Analysis of Building Using Line Segments and Their Geometrical Vanishing Points. In: Proceeding of SICE-ICASE (2006)
29. Trinh, H.H., Kim, D.N., Jo, K.H.: Facet-based Multiple Building Analysis for Robot Intelligence. *Journal of Applied Mathematics and Computation (AMC)* 205(2), 537–549 (2008)
30. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM* 395, 381–395 (1981)
31. Harris, C., Stephens, M.: A combined corner and edge detector, in Proceedings of the 4th Alvey Vision Conference, Manchester, UK, pp. 147–151 (1998)
32. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
33. Juan, L., Gwun, O.: A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing* 3(5) (2010)
34. Jean-Yves, Bouguet: Camera Calibration Toolbox for Matlab,
http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

Design of an Energy Consumption Scheduler Based on Genetic Algorithms in the Smart Grid

Junghoon Lee¹, Gyung-Leen Park¹, Ho-Young Kwak², and Hongbeom Jeon³

¹ Dept. of Computer Science and Statistics, Jeju National University

² Dept. of Computer Engineering, Jeju National University

³ Smart Green Development Center, KT, Republic of Korea

Abstract. This paper designs an energy consumption scheduler capable of reducing peak power load in smart places based on genetic algorithms and measures its performance. The proposed scheme follows the task model consisting of actuation time, operation length, deadline, and a consumption profile, while each task can be either nonpreemptive or preemptive. Each schedule is encoded to a gene, each element of which element represents the start time for nonpreemptive tasks and the precalculated combination index for preemptive tasks. The evolution process includes random initialization, Roulette Wheel selection, uniform crossover, and replacement for duplicated genes. The performance measurement result, obtained from a prototype implementation of both the proposed genetic scheduler and the backtracking-based optimal scheduler, shows that the proposed scheme can always meet the time constraint of each task and keeps the accuracy loss below 4.7 %, even for quite a large search space. It also achieves uncomparable execution time of just a few seconds, which makes it appropriate in the real-world deployment.

Keywords: Smart grid, power consumption scheduler, genetic algorithm, combinatory index, peak load reduction.

1 Introduction

It is undeniable that the reliable and efficient supply of electric power is essential in our everyday life and the modern power system is taking advantage of information technology to make itself smarter and more intelligent [1]. The new operating strategies, called the smart grid, innovates legacy power systems, especially in power system management and intelligent load control. The smart grid also embraces a variety of energy sources such as solar, wind, and other renewable energies. From the viewpoint of customers, the smart grid saves energy, reduces cost, and improves reliability. Moreover, they can smartly consume electricity by selecting the preferred supplier via the two-way interaction between the two parties [2]. Many countries are necessarily interested in this smart grid system, trying to take the initiative in its research, technique, and business.

In the mean time, the Republic of Korea was designated as one of the smart grid initiative countries together with Italy during the expanded G8 Summit in

2009, while its roadmap is included in 10 transformative technologies reducing greenhouse emissions [3]. Korea is now shifting toward a low carbon economy and a society capable of recovering from climate change. In light of this, Korea launched a proactive and ambitious plan to build a smart grid test-bed on Jeju Island to prove its determination in the green-growth strategy in June 2009. Jeju island is located at the southernmost tip of the country, particularly having abundant wind energy. The Jeju test-bed will become the world's largest smart grid community that allows testing of the most advanced smart grid technologies and R&D results, as well as the development of business models. Particularly, a total of about 60 million USD will be invested between 2009 and 2013. Currently, 9 consortiums in five areas are participating in the enterprise.

In the smart grid, the role of information and automation technologies increases extensively to make smarter not only the power distribution network but also the demand-side management [4]. An enterprise-level information system includes supervisory control, data acquisition, customer services, planning, trading, scheduling, power marketing, billing, accounting, and business management [5]. Out of a lot of benefits obtainable from the smart grid, peak power reduction is very important not just in economic but also environmental aspects. The peak hour may lead to the temporary employment of expensive dispatchable energy. Moreover, if the power consumption exceeds the current capacity of power transmission cable in each home, building, farm, and so on, catastrophic power outage can take place, making it necessary to rebuild the cable system [6]. Moreover, the area-wide power shortage makes more power plants be built.

The peak load reduction can be achieved by the load control in a microgrid, where a controller device schedules the operation of each appliance according to its power requirement, current pricing policy, and accumulated power availability [7]. As an essential function block of DR (Demand Response), the load controller can reshape the power load by scheduling the operation of each electric device [4]. However, scheduling is in most cases a very complex time-consuming problem greatly sensitive to the number of tasks. It is difficult to solve by conventional optimization schemes, and their severe execution time makes them hard to apply in our daily life. In this regard, this paper is to design a suboptimal but practical appliance scheduler capable of reducing peak power consumption in homes or buildings, namely, smart places [8], based on genetic algorithms and measure its performance by means of the prototype implementation. Our design focuses on how to encode the appliance schedule, how to apply genetic operations, and how much peak load reduction we can achieve.

2 Background and Related Work

For the first example of power management, MAHAS (Multi-Agent Home Automation System) can adapt power consumption to available power resources according to inhabitant comfort and cost criteria [9]. Based on the multi-agent architecture, the power management problem is divided into subproblems involving different agents, each of which tries to solve its own problem independently

to find a solution of the whole problem. Particularly, the control algorithm is decomposed into reaction and anticipation mechanisms. While the first protects constraint violations, the second computes the plan for global consumption according to predicted productions and consumptions. The control function coordinates and negotiates the agent operations, sometimes eliminating or adding new agents. While this scheme can reduce down the whole search space for the given optimization problem, it cannot guarantee obtaining the optimal solution or avoid the complex interaction between the agents.

[10] discusses a scheduling problem for household tasks to help users save money spent on their energy consumption. Assuming the situation customers are offered with options to select their preferred electricity supplier company, its system model relies on electricity price signals, availability of locally generated power, and flexible tasks with deadlines. Noticeably, this work presents a descriptive task model where tasks are either preemptive or nonpreemptive. Our paper also follows the same task model. A case study shows that cost savings are possible, but fast and efficient solutions are still needed. This problem stems from the fact that they use relatively fine grained time slots, as large as 20 minutes, not employing any heuristic. So, its time complexity reaches $O(2^{MN})$, where N is the number of tasks and M is the number of time slots.

The small autonomous hybrid power system, or SAHPS in short, contains renewable and conventional power sources. Conventional generators produce on demand in economic way, and they can provide backup power when the renewable production is not sufficient. Even though the renewable energy source, combined with electric storage, does not emit during their operation, they may produce significant amount of pollutant emissions in their whole life cycle. In its design, economic and environmental criteria are two conflicting objectives. [11] proposes that the economic objective function should be system's cost of energy, while the environmental one is total CO₂ emissions. Specifically, non-determinant sorting genetic algorithm is combined with a local search procedure to solve the multi-objective optimization problem. In addition, [12] designs a hybrid PV-wind-diesel-hydrogen-battery installation for the generation of electric energy considering three conflicting objectives, namely, cost, pollution, and unmet load) based on evolutionary algorithm.

In addition, our previous work has designed a power management scheme capable of reducing the peak power consumption [7]. It finds the optimal schedule for the task set consisting of nonpreemptive and preemptive tasks, each of which has its own consumption profile as in [10]. To deal with the intolerable scheduling latency for the case of large number of tasks and slots, two speed enhancement techniques are integrated. First, for a nonpreemptive task, the profile entries are linearly copied into the allocation table without intermittence. Second, for the preemptive task, the feasible combinatory allocations are generated and stored in advance of search space expansion. Then, for every partial allocation just consist of nonpreemptive tasks, the scheduler maps the combination of each preemptive task to the allocation table one by one, checking the peak power requirement. This scheme reduces the scheduling time almost to 2 %, compared with [10].

3 Task Scheduler

3.1 System Model

Figure 1 illustrates our system model. Even though the figure shows a home management system, it can be extended to buildings, farms, sensor & actuator networks, and the like. Basically, the smart home includes an in-home network and a WAN connection. The in-home network connects the power control system, in-home display, or IHD in short, and possibly other computing devices such as PC. The IHD provides the user interface for customers to order the operation of specific appliances. The IHD or PC may generate or modify the appliance schedule according to changes in the task set or price signal. The schedule is then sent to the power control system which has power line interconnection to each of electric devices to control their operations. There exist a set of available home area networks such as Zigbee and power line communication [14].

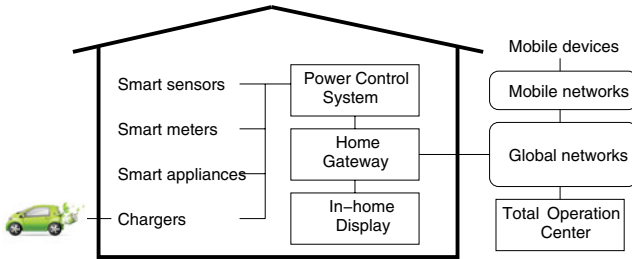


Fig. 1. System model

The home gateway connects in-home devices to global services, allowing bidirectional interaction between a consumer and a provider. The utility company sends a residential load change to the consumer's home, activating load control, demand response, and price adjustment. In addition, the smart grid home controller sends a message to a human user through the mobile phone or web portal to inform him or her of specific events and price changes. It must be mentioned that the scheduler can reside not only in an in-home device but also in the mobile phone, TOC (Total Operation Center), and other remote high-performance servers. If the scheduler is implemented in TOC, it is possible to globally coordinate the power schedule to avoid the peak resonance in the multiple microgrids which employ the same scheduling algorithm.

3.2 Task Model

Each task has its own power consumption characteristics. Some tasks must start immediately after they get ready and cannot be ceased to the end. For example, a hair drier starts to run at the moment a user wants. Such inflexible tasks are not

schedulable as there is no other option to adjust their start time. So, we will not consider them in the schedule generation. For schedulable tasks, there are two classes, namely, preemptive and nonpreemptive tasks. The two classes commonly have task deadlines. For example, the task must be completed before a customer leaves for his office or before he returns home. A dish washer or a laundry machine can start any time as long as the task can be completed within a specific deadline. As its operation cannot be suspended, it belongs to the nonpreemptive task. As contrast, the electric car charge belongs to the preemptive task as its operation can be suspended and resumed within its deadline. After all, Task T_i can be modeled with the tuple of $\langle F_i, A_i, D_i, U_i \rangle$. F_i indicates whether T_i is preemptive or nonpreemptive. A_i is the activation time of T_i , D_i is the deadline, and U_i denotes the operation length.

The load power profile is practical for characterizing the power consumption behavior of each appliance. As pointed in [10], the load power profile for a washing machine depends on the set program, its duration, and the water temperature chosen by the user. The scheduler can assume that the power requirement of each operation step is known in priori. Here, the power consumption pattern for every electric device is aligned to the fixed-size time slot. Actually, each device has its own time scale in its power consumption. However, we can take the average value during each time slot considering AVR (Automatic Voltage Regulator) or UPS (Uninterruptible Power Supply). The length of a time slot can be tuned according to the system requirement on the schedule granularity and the computing time. Figure 2(a) plots the sample power consumption profile for 6 tasks, where Task 5 and Task 6 are preemptive. In this figure, we do not specify the power unit, as the power scale depends on the specific electric device types.

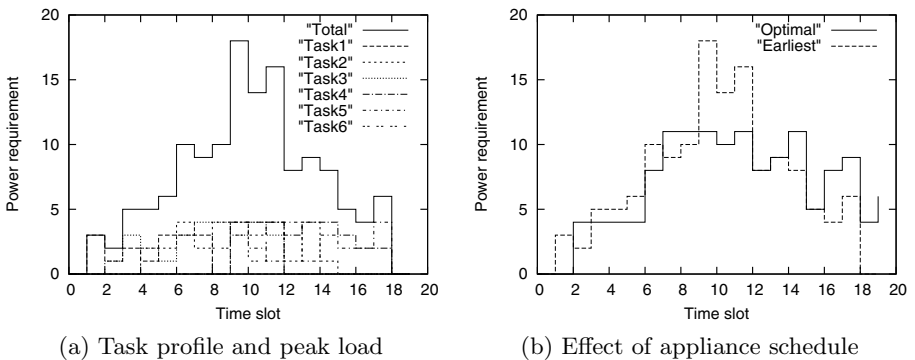


Fig. 2. Effect of appliance schedule

The operation of each appliance has to be completed within a deadline. Hence, the appliance schedule is quite similar to the real-time task schedule. However, the appliance schedule must further consider the peak power reduction in time slots. The scheduling problem leads to the allocation of each device operation

to the $N \times M$ allocation table, where N is the number of tasks and M is the number of time slots. A nonpreemptive task can start from its activation time to the latest start time, which can be calculated by subtracting U_i from D_i . In the task set shown in Figure 2(a), Task 1 can be started at any time slot from 2 to 12 to meet its time constraint. As contrast, the preemptive task case is quite complex. To meet its time constraint, the task must run for U_i out of $(D_i - A_i)$ slots, so the number of feasible options is equal to ${}_{(D_i - A_i)}C_{U_i}$.

Without scheduling, the peak power can reach 19 at time slot 9 as shown in Figure 2(a), which plots the per-slot power requirement with the solid line. Figure 2(b) shows that the peak power can be reduced to 11 by an optimal schedule, demonstrating the advantage of power scheduling. The optimal solution is obtained by the backtracking-based search space traversal which checks all feasible solutions [7]. However, it takes tens of minutes or sometimes a couple of hours in the average performance PC to generate an optimal schedule even with constraint checking, when the number of tasks gets larger than 6. The execution time is more affected by the number of nonpreemptive tasks. So, the backtracking-based search is impractical in spite of the optimality it can guarantee.

3.3 Proposed Scheme

Genetic algorithms are efficient search techniques based on principles of natural selection and genetics. They have been successfully applied to find acceptable solutions to problems in business, engineering, and science within a reasonable time amount [13]. Each evolutionary step generates a population of candidate solutions and evaluates the population according to a fitness function to select the best solution and mate to form the next generation. Over a number of generations, good traits dominate the population, resulting in an increase in the quality of the solutions. It must be mentioned that the genetic algorithm process can run for years and does not find any better solution than it did in the first part of the process.

In the scheduling problem, a chromosome corresponds to a single feasible schedule, and is represented by a fixed-length string of integer-valued vector. A value element denotes the start time for nonpreemptive tasks. As they cannot be suspended once they have begun, just the start time is enough to describe their behaviors in a schedule. Here, if the consumption profile of the task is (3, 4, 5, 2), and the value is 2, the allocation for this task will be (0, 0, 3, 4, 5, 2, 0, 0, ...). As contrast, for a preemptive task, possible slot allocations are generated in advance and numbered as shown in Figure 3. In this example, $D_i - A_i$ is 5 and U_i is 3, so the number of possible allocations is ${}_5C_3$. Each of them is numbered from 0 to 9. Hence, if the value in the vector is 1, the mapping vector is (0, 1, 0, 1, 1) and each profile entry is mapped to the position having 1 one by one from the start time, namely, 16. Hence, the allocation will be (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 3, 0, 4). The allocation vector can be converted into the allocation table which has N rows and M columns. For each allocation, the scheduler can calculate the per-slot power requirement and the peak load.

$A_i = 15 \quad D_i = 20 \quad U_i = 3 \quad \text{Profile (2, 3, 4)}$			
Index	Combination	Index	Combination
0	0 0 1 1 1	5	1 0 1 0 1
1	0 1 0 1 1	6	1 0 1 1 0
2	0 1 1 0 1	7	1 1 0 0 1
3	0 1 1 1 0	8	1 1 0 1 0
4	1 0 0 1 1	9	1 1 1 0 0

Fig. 3. Preemptive task combination index

The iteration consists of selection and reproduction. Selection is a method that picks parents according to the fitness function. The Roulette Wheel selection gives more chances to genes having better fitness values for mating. Reproduction, or crossover, is the process taking two parents and producing a child with the hope that the child will be a better solution. This operation randomly selects a pair of two crossover points and swaps the substrings from each parent. Reproduction may generate the same gene with the existing ones in the population. It is meaningless to have multiple instances of a single schedule. So, they will be replaced by new random genes. Additionally, mutation exchanges two elements in a gene. In our scheme, the meaning of the value is different for preemptive and nonpreemptive tasks. Hence, the mutation must be prohibited. The appliance scheduler is subject to time constraint. However this constraint can be always met, as the scheduler selects the start time only within the valid range and the precalculated combination index.

4 Performance Measurement

This section implements the proposed allocation method using Visual C++ 6.0, making it run on the platform equipped with Intel Core2 Duo CPU, 3.0 GB memory, and Windows Vista operating system. The experiment sets the schedule length, namely, M , to 20 time units. If a single time unit is 20 min as in [10], the total schedule length will be 6.6 hours, and it is sufficiently large for the customer appliance schedule. For a task, the start time and the operation time are selected randomly between 0 and $M - 1$, but it will be set to $M - 1$ if the finish time, namely, the sum of start time and the operation length, exceeds M . All tasks have the common deadline, namely, M , considering the situation that all tasks must be done before the office hour begins or ends. In addition, the power level for each time slot has the value of 1 through 5. As this paper aims at enhancing the computation speed, the performance measurement concentrates on how much accuracy is lost and how much speed-up can be obtained, comparing with an optimal schedule implementation [7].

First, Figure 4(a) plots the operation of the genetic scheduler, highlighting the stepwise reduction in peak power consumption according to the iteration for the task set shown in Figure 2(a). In the graph, two reference lines denote

the peak values for the earliest schedule taking no specific scheduling strategy and the optimal schedule obtained by the backtracking-based search. The search space traversal takes almost 30 minutes as the task set contains two preemptive tasks. The peak load for the genetic scheduler begins from 15 with the random selection of start time and combination index for the initial population. As the evolution step proceeds, the peak value reaches to the optimal value, namely, 10. In this case, the scheduler gets the optimal solution just in 250 iterations, which is less than 1 second.

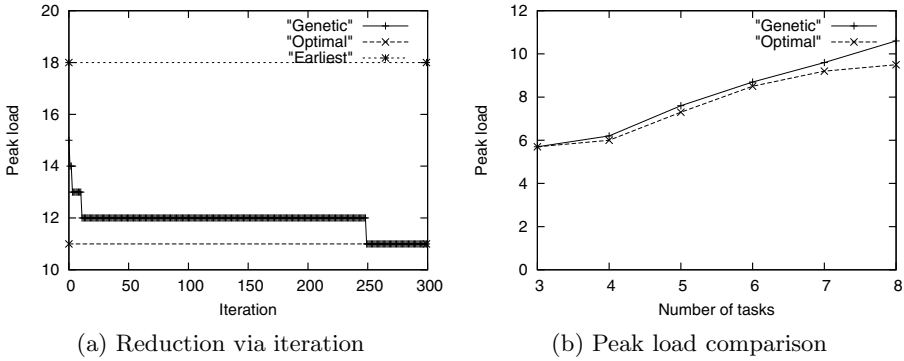


Fig. 4. Iterative reduction and peak load

The second experiment measures the peak load reduction according to the number of tasks ranging from 3 to 8, while the number of preemptive tasks is set to 2 for all task sets. For each task group having the same number of tasks, 10 sets are generated. The experiment runs both the genetic scheduler and the optimal scheduler. Then, the maximum power values, namely, peak load of respective sets are measured and averaged. The genetic scheduler limits the number of iterations to 1000, which corresponds to about 10 seconds for 8 tasks. The number of genes in a population is 20. Figure 4(b) compares the peak loads obtained by two schemes. The difference gap tends to get larger according to the increase of the number of tasks, reaching 10.3 % when the number of tasks is 8. Up to 7 tasks, the gap remains below 4 %. The experiment result indicates that the proposed scheme can generate an acceptable solution comparable to the optimal schedule. The execution time of the optimal scheduler approaches to couple of hours quite often. Hence, the execution time comparison is meaningless.

The next experiment measures the effect of iterations in finding a better schedule in terms of accuracy. The experiment runs the genetic and optimal schedulers for all task sets used in the previous experiment. We define the accuracy for the schedule as the ratio of the peak load in the genetic schedule to that in the optimal schedule. If the accuracy is 1.0, the schedule is an optimal one. Figure 5(a) plots the accuracy of schedules generated by our scheme. As can be seen in the figure, accuracy improves according to the increase of iteration steps, that is, as the difference between two schemes decreases. However, after 600 steps,

the improvement is not so significant. Actually, our observation finds out that the peak load reduction obtained before 100 iterations hardly gets improved in subsequent interactions. We think that it's because the preemptive part schedule can take smaller options, compared with the nonpreemptive part.

Finally, Figure 5(b) plots the optimality of the proposed scheme. We define the optimality as the probability that a scheduling scheme can find the optimal schedule for a given task set. In addition to accuracy in the peak power, it is a useful performance metric to evaluate the efficiency of a scheduling scheme. Our scheduler finds the optimal schedule for all 10 sets when the number of tasks is 3. However, when it is 8, just 1 out of 10 sets succeeds in finding the optimal schedule. Until the number of tasks is 7, the optimality remains above 0.6. After all, these results show that the proposed scheduler can find an acceptable schedule within 10 seconds, uncomparably smaller than the backtracking-based scheduler. The worst peak load difference is 10.3 % for the given parameter set.

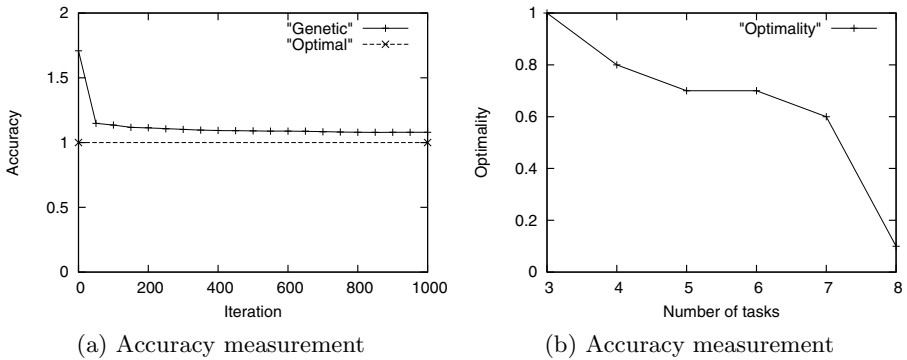


Fig. 5. Accuracy analysis

5 Conclusions

This paper has presented a design and measured the performance of a suboptimal appliance scheduler capable of reducing peak power consumption in a microgrid and meeting the time constraint of each appliance operation. To overcome the excessive execution time of the optimal scheduling schemes such as backtracking-based search tree traversal, our scheme has designed a scheduler based on genetic algorithms, which can obtain an acceptable schedule within a small time bound. The proposed scheme follows the task model consisting of actuation time, operation length, deadline, and a consumption profile, while each task can be either nonpreemptive or preemptive. Each schedule is encoded to a gene based on the start time for the nonpreemptive task and on the precalculated combination index for the preemptive task. The evolution process includes random initialization, Roulette Wheel selection, uniform crossover, and replacement for duplicated genes. The performance measurement has been conducted through

the prototype implementation of the genetic scheduler and the backtracking-based optimal scheduler. The experiment result shows that the proposed scheme can always meet the time constraint of each appliance operation and keep the accuracy loss below 4.7 % in most cases. As it takes just a few second to get a schedule, it can be deployed in the real-world product and can deal with more tasks.

Acknowledgments. This research was supported by the MKE (The Ministry of Knowledge Economy), through the project of Region technical renovation, Republic of Korea.

References

1. Gellings, C.: *The Smart Grid: Enabling Energy Efficiency and Demand Response*. The Fairmont Press (2009)
2. Al-Agtash, S., Al-Fahoum, A.: An evolutionary computation approach to electricity trade negotiation. In: *Advances in Engineering Software*, pp. 173–179 (2005)
3. <http://www.smartgrid.or.kr/10eng3-1.php>
4. Spees, K., Lave, L.: Demand response and electricity market efficiency. *The Electricity Journal*, 69–85 (2007)
5. Ipakchi, A., Albuyeh, F.: Grid of the future. *IEEE Power & Energy Magazine*, 52–62 (2009)
6. Facchinetti, T., Bibi, E., Bertogna, M.: Reducing the peak power through real-time scheduling techniques in cyber-physical energy systems. In: *First International Workshop on Energy Aware Design and Analysis of Cyber Physical Systems* (2010)
7. Lee, J., Park, G., Kim, S., Kim, H., Sung, C.: Power consumption scheduling for peak load reduction in smart grid homes. In: *To Appear at Symposium on Applied Computing* (2011)
8. Mady, A., Boubekeur, M., Provan, G.: Optimised embedded distributed controller for automated lighting systems. In: *First Workshop on Green and Smart Embedded System Technology: Infrastructures, Methods, and Tools* (2010)
9. Abras, S., Pesty, S., Ploix, S., Jacomino, M.: An anticipation mechanism for power management in a smart home using multi-agent systems. In: *3rd International Conference on From Theory to Applications*, pp. 1–6 (2008)
10. Derin, O., Ferrante, A.: Scheduling energy consumption with local renewable micro-generation and dynamic electricity prices. In: *First Workshop on Green and Smart Embedded System Technology: Infrastructures, Methods, and Tools* (2010)
11. Katsigiannis, Y., Georgilakis, P., Karapidakis, E.: Multiobjective genetic algorithm solution to the optimum economic and environmental performance problem of small autonomous hybrid power systems with renewables. *IET Renewable Power Generation*, 404–419 (2010)
12. Dufo-Lopez, R., Bernal-Agustin, J.: Multi-objective design of PV-wind-diesel-hydrogen-battery systems. *Renewable Energy*, 2559–2572 (2008)
13. Cantu-Paz, E.: *A survey of parallel genetic algorithms*. *Calculateurs Paralleles* (1998)
14. Gislason, D.: *ZIGBEE Wireless Networking*, Newnes (2008)

Toward Cyclic Scheduling of Concurrent Multimodal Processes

Grzegorz Bocewicz¹, Robert Wójcik², and Zbigniew A. Banaszak³

¹ Koszalin University of Technology,
Dept. of Computer Science and Management, Koszalin, Poland
bocewicz@ie.tu.koszalin.pl

² Wrocław University of Technology, Institute of Computer Engineering,
Control and Robotics, Wrocław, Poland
robert.wojcik@pwr.wroc.pl

³ Warsaw University of Technology, Faculty of Management,
Dept. of Business Informatics, Warsaw, Poland
Z.Banaszak@wz.pw.edu.pl

Abstract. The problem of cyclic scheduling of multimodal cyclic processes (MCPs) is considered. The issue follows the production engineering and supply chains environment, where the imposition of the integer domain results (due to inherent process features such as discrete slot sizes, etc.) in the Diophantine character of a scheduling problem. Consequently, some classes of MCPs scheduling problems can be regarded as non-decidable ones. Since system constraints its behavior, both system structure configuration and desired schedule have to be considered simultaneously. Therefore, MCP scheduling problem solution requires that the system structure configuration must be determined for the purpose of processes scheduling, yet scheduling must be done to devise the system configuration. The approach proposed in this paper provides the framework allowing one to take into account both direct and reverse formulation of the cyclic scheduling problem. It permits to determine the model for the assessment of the impact of the structure of local cyclic processes on the parameters of global MCP. Discussion of some solubility issues concerning multimodal cyclic process dispatching problems is provided.

Keywords. Cyclic processes, concurrent processes, scheduling, Diophantine problem, state space, multimodal processes.

1 Introduction

A cyclic scheduling problem is a scheduling problem in which some set of activities is to be repeated an indefinite number of times, and it is desired that the sequence be repeating. Cyclic scheduling problems arise in different application domains (such as manufacturing, supply chains, time-sharing of processors) as well as service domains (covering such areas as workforce scheduling, train timetabling, aircraft routing, and reservations) [4], [5], [6]. The scheduling problems considered in this paper belong to the class of NP-hard ones and are usually formulated in terms of decision problems,

i.e. in terms of searching for an answer whether a solution having required features exists or not [9]. More formally, a decision problem can be addressed as a question stated in some formal system with a yes-or-no answer, depending on the values of some input parameters. The decision problems, in turn, fall into two categories: decidable and non decidable problems, [8], [11], [12].

The rest of the paper is organized as follows: Section 2 describes the class of systems composed of concurrently flowing cyclic processes (SCCPs). The SCCPs treated in terms of Diophantine problems are then formally modelled in Section 3 where some issues regarding a state space and cyclic steady state reachability are investigated. In Section 4 a cyclic scheduling problem of multimodal concurrently flowing cyclic processes is formulated and briefly illustrated. Conclusions are presented in Section 5.

2 Systems of Concurrent Cyclic Processes

Consider a system of repetitive manufacturing processes sharing common resources while following a distributed mutual exclusion protocol (see Fig. 1).

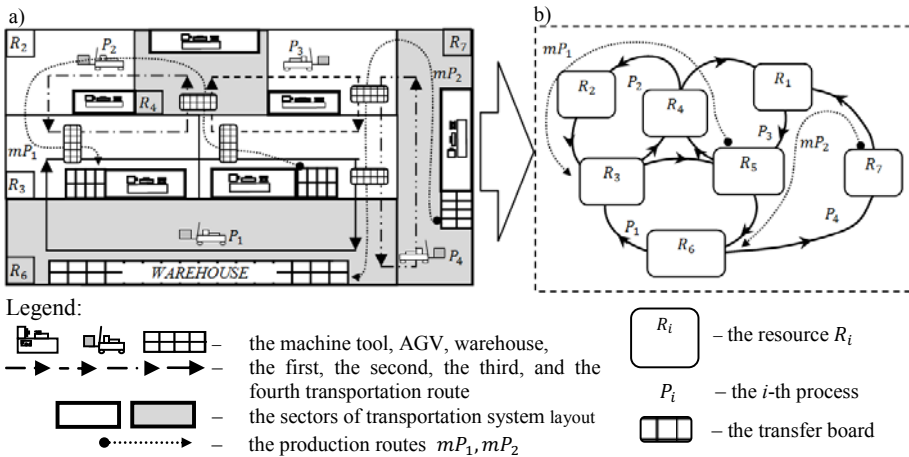


Fig. 1. An AGV system modeled in terms of concurrent cyclic processes

Each process P_i ($i = 1, 2, \dots, n$), representing one product processing, executes periodically a sequence of the operations employing one resources defined by $p_i = (R_{j_1}, \dots, R_{j_{lp(i)}})$, $j_k \in \{1, 2, \dots, m\}$, where $lp(i)$ denotes a length of production route and m denotes number of resources, i.e., $R_{j_k} \in R$, where R is a set of resources of SCCP: $R = \{R_1, R_2, \dots, R_m\}$. The time $t_{i,j} \in \mathbb{N}$, of operation executed on R_j along P_i , is defined in domain of uniform time units (\mathbb{N} – the set of natural numbers). The sequence $T_i = (t_{i,j_1}, \dots, t_{i,j_{lp(i)}})$ of operation times follows P_i . In case shown in Fig.1 the SCCP consists of 7 resources and 4 processes. The resources R_1, R_3, R_4, R_5, R_6 , are shared ones, since each one is used by at least two processes, and the resources R_2, R_7 , are non-shared since each one is exclusively used by only one process. The

processes P_1, P_2, P_3, P_4 execute along the transportation routes given by sequences: $p_1 = (R_3, R_5, R_6)$, $p_2 = (R_2, R_4, R_3)$, $p_3 = (R_1, R_5, R_4)$, $p_4 = (R_7, R_1, R_5, R_6)$, respectively, and two MCPs mp_1, mp_2 are executed along routes $mp = \{mp_1, mp_2\}$ which are sequences of sub sequences (parts) of local cyclic processes, i.e., P_1, P_2, P_3, P_4 .

A frequently faced question is whether a production order can be completed in a required time period? Such question is usually considered under the assumption that some production capacities are still available in the enterprise. The issue is whether the production order can be accepted for execution when the availability of some resources is constrained in time. Other questions considered in process scheduling design are [2], [3]: Does there exist such an initial processes allocation that leads to a steady state in which no process waits to the access to the common shared resources? What set of priority dispatching rules assigned to shared resources guarantee, if either, the same rate of resources utilization? So, in general case the questions may regard to the qualitative features of system behavior such as deadlock and/or conflicts avoidance [7], [10]. For example they may be aimed at conditions satisfaction of which guarantees system repetitiveness for a given initial state and/or the dispatching rules allocation.

In order to illustrate the Diophantine character of the cyclic processes scheduling, let us consider the system of concurrently flowing cyclic processes shown in Fig. 2.

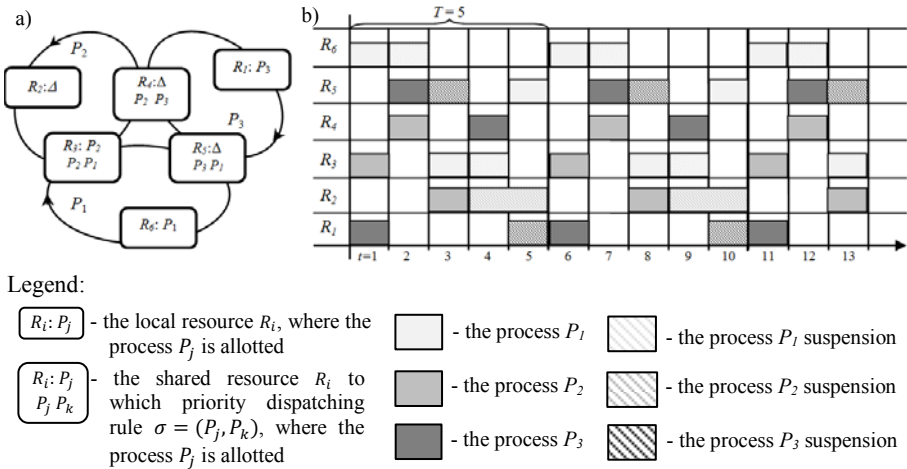


Fig. 2. System of concurrently flowing cyclic processes: a) initial state, b) Gantt's chart encompassing the cyclic steady state

At the initial state (see Fig. 2 a)) the steady state cyclic system behavior, illustrated by Gantt's chart (see Fig. 2 b)), is characterized by periodicity $T = 5$ (obtained under assumption $t_{1,3} = t_{1,5} = t_{1,6} = t_{2,3} = t_{2,2} = t_{2,4} = t_{3,4} = t_{3,1} = t_{3,5} = 1$).

Note that besides of the initial processes allocation (see the sequence $A_0 = (R_1, R_3, R_6)$) the cyclic steady state behavior depends on the routings direction as well as the priority rules that determine the order in which processes make their

access to the common shared resources. For instance changing the priority rules into $\sigma_3 = (P_2, P_1)$, $\sigma_4 = (P_2, P_3)$, $\sigma_5 = (P_3, P_1)$ is guaranteeing the cyclic steady state behavior, and if $\sigma_3 = (P_2, P_1)$, $\sigma_4 = (P_3, P_2)$, $\sigma_5 = (P_1, P_3)$, then the resultant state is a deadlock one.

The periodicities of the cyclic steady states can be calculated from the linear Diophantine equation obtained assuming that:

- the initial state and set of dispatching rules guarantee there exists admissible solution (i.e. cyclic steady state)
- the graph model of concurrent processes is consistent.

Consider the following set of equations:

$$x_{P1} \cdot (t_{1,3} + t_{1,5} + t_{1,6}) + y_{P2} \cdot t_{2,3} + z_{P3} \cdot t_{3,5} = Tc \quad (1)$$

$$y_{P2} \cdot (t_{2,3} + t_{2,2} + t_{2,4}) + x_{P1} \cdot t_{1,3} + z_{P3} \cdot t_{3,4} = Tc \quad (2)$$

$$z_{P3} \cdot (t_{3,4} + t_{3,1} + t_{3,5}) + x_{P1} \cdot t_{1,5} + y_{P2} \cdot t_{2,4} = Tc \quad (3)$$

where: $t_{i,j}$ – the execution time of the operations performed on the j -th resource along the i -th process, T – the periodicity of the system of concurrently executed cyclic processes, x_{P1} , y_{P2} , z_{P3} – the number of times the process P_i repeats within the period Tc .

Subtracting equation (3) from equation (2) the resulting equation has the form:

$$y_{P2} \cdot t_{2,3} + y_{P2} \cdot t_{2,2} + x_{P1} \cdot t_{1,3} - z_{P3} \cdot t_{3,1} - z_{P3} \cdot t_{3,5} - x_{P1} \cdot t_{1,5} = 0 \quad (4)$$

After combining (1) with (4), the resultant formula has the form:

$$y_{P2} \cdot (2 \cdot t_{2,3} + t_{2,2}) + x_{P1} \cdot (2 \cdot t_{1,3} + t_{1,6}) - z_{P3} \cdot t_{3,1} = Tc. \quad (5)$$

Consequently, $y_{P2} \cdot N + x_{P1} \cdot M = Tc + z_{P3} \cdot K$ which is a Diophantine equation, where $N = M = 3$ and $K = 1$ subject to $t_{1,3} = t_{1,5} = t_{1,6} = t_{2,3} = t_{2,2} = t_{2,4} = t_{3,4} = t_{3,1} = t_{3,5} = 1$, y_{P2} , x_{P1} , $z_{P3} \in \mathbb{N}$, results in the following form: $3y_{P2} + 3x_{P1} = Tc + z_{P3}$. Finally, since dispatching rules assumed results in $x_{P1} = y_{P2} = z_{P3}$, i.e. different processes P_i repeat same number of times within the period Tc , hence $Tc \in \{5, 10, 15, 20, \dots\}$.

The straight and reverse problems can be considered. In the first case, assuming the operation time t_{ij} , the response to the following question is sought: Does the system periodicity can be equal to assumed value Tc ? In other words, the question concerns of x_{P1} , y_{P2} , z_{P3} following $y_{P2} \cdot N + x_{P1} \cdot M = Tc + z_{P3} \cdot K$.

In the second case, assuming x_{P1} , y_{P2} , z_{P3} , the response to the following question is sought: Does there exist a system structure guaranteeing cyclic process execution with period Tc ? In other words, the question concerns the values of N , M and K satisfying $y_{P2} \cdot N + x_{P1} \cdot M = Tc + z_{P3} \cdot K$.

Assuming the set of variables (including dispatching rules and initial processes allocation) and Diophantine equations modeling the system structure, while the set of solutions following given system features (its periodicity, the number of times the local process repeats within the period Tc , etc.), one can pose the following questions:

Does there exist a control procedure that guarantees an assumed system behavior subject to system structure constraints? Does there exist a system structure such that an assumed system behavior can be achieved?

Therefore, taking into account non decidability of Diophantine problems, one can easily realize that not all behaviors can be obtained under constraints imposed by the system structure. The similar observation concerns the system behavior that can be achieved in systems with specific structural characteristics.

3 Cyclic Steady State Reachability Problem

Consider the Systems of Concurrent Cyclic Processes (SCCP) as in Fig. 3 a) (which represents SCCP from Fig. 1) specified by the structure (ST, TO) , where

$$ST = \{(R_6, R_3, R_5), (R_2, R_3, R_4), (R_1, R_5, R_4), (R_7, R_1, R_5, R_6)\}, \tag{6}$$

$$TO = \{t_{1,6}, t_{1,3}, t_{1,5}, \dots, t_{4,6}\}, t_{i,j} = 1, \forall t_{i,j} \in TO, \tag{7}$$

and sets of decision variables $\{\Theta, \mathbb{S}\}$, where

- Θ – the family of possible dispatching rules; in the case considered $\Theta \in \mathbb{O}$, $\Theta = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7\}$,
 where: $\sigma_1 = (P_3, P_4)$, $\sigma_2 = (P_2)$, $\sigma_3 = (P_2, P_1)$, $\sigma_4 = (P_3, P_2)$, $\sigma_5 = (P_3, P_4, P_1)$, $\sigma_6 = (P_1, P_4)$, $\sigma_7 = (P_4)$
- \mathbb{S} – the set of admissible states; in the case considered $S^0 \in \mathbb{S}$, $S^0 = (A^0, Z^0)$,
 where: $A^0 = (P_3, \Delta, P_2, \Delta, \Delta, P_1, P_4)$ is an initial processes allocation, and $Z^0 = (P_3, P_2, P_2, P_2, P_3, P_1, P_4)$ is an initial semaphore.

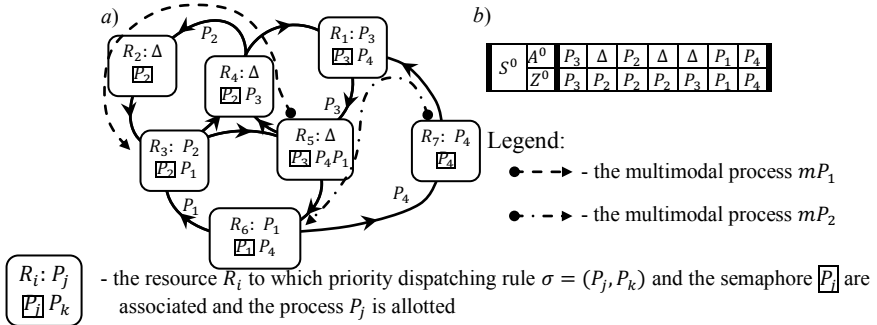


Fig. 3. Representations of the state: snapshots-like a), table-like b)

In order to simplify further discussion, let us assume the following table-like representation of an initial state (see Fig. 3 b). So, in the case considered here, the process P_1 is allocated to the resource R_6 , the process P_2 to the resource R_3 , the process P_3 to R_1 and the process P_4 to R_7 ; while due to the semaphore assumed, the process P_2 may be allocated on R_2 or R_3 or R_4 , while P_3 either on R_1 or R_5 , and P_2 and P_4 on R_6 and R_7 , respectively.

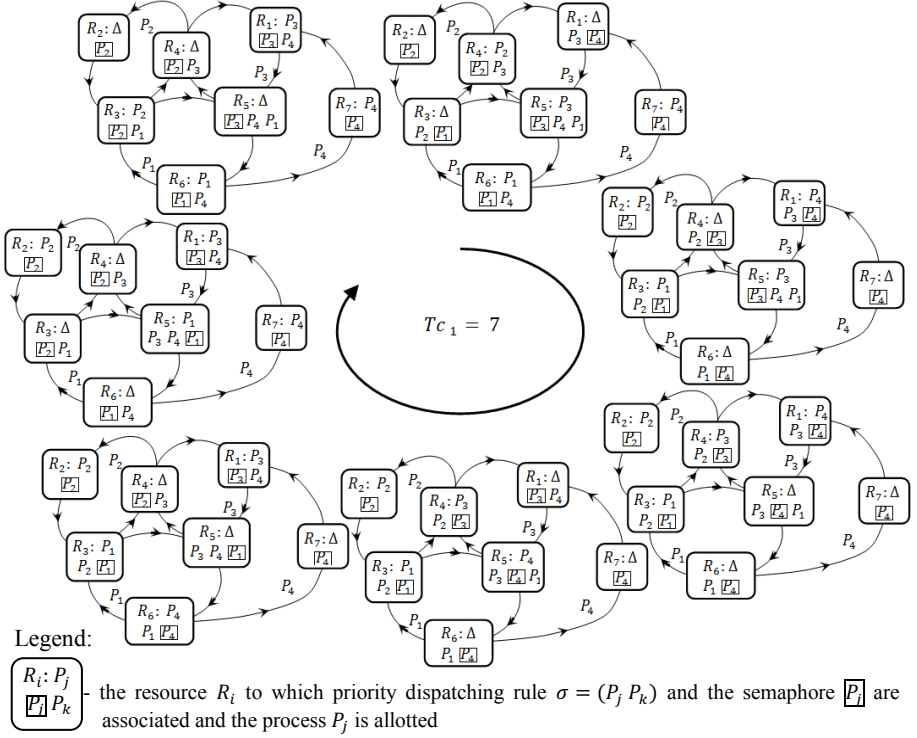


Fig. 4. Snapshot-like illustration of the cyclic steady state

Consequently, the following quintuple can be treated as a model of SCCP,

$$(ST, TO, \Theta, S, \delta) \quad (8)$$

where δ – is the next state function following the conditions below:

$$\begin{aligned} & \forall_{i \in \{1, 2, \dots, m\}} \forall_{j \in \{1, 2, \dots, n\}} [(a_i^k = \Delta) \wedge (a_{\beta_i(P_j)}^k = z_i^k) \Rightarrow (a_i^{k+1} = z_i^k)], \\ & \forall_{i \in \{1, 2, \dots, m\}} \forall_{j \in \{1, 2, \dots, n\}} [(a_i^k = \Delta) \wedge (a_{\beta_i(P_j)}^k \neq z_i^k) \Rightarrow (a_i^{k+1} \neq P_j)], \\ & \forall_{i \in \{1, 2, \dots, m\}} [(a_i^k = \Delta) \Rightarrow (z_i^{k+1} = z_i^k)], \\ & \forall_{i \in \{1, 2, \dots, m\}} [(a_i^k \neq \Delta) \wedge (a_i^{k+1} \neq \Delta) \Rightarrow [(z_i^{k+1} = z_i^k) \wedge (a_i^{k+1} = a_i^k)]], \\ & \forall_{i \in \{1, 2, \dots, m\}} [(a_i^k \neq \Delta) \wedge (a_i^{k+1} = \Delta) \Rightarrow (z_i^{k+1} = \varphi_i(z_i^k))], \\ & \forall_{i \in \{1, 2, \dots, m\}} [(a_i^k \neq \Delta) \wedge (z_{\alpha_i(a_i^k)}^k = a_i^k) \Rightarrow (a_{\alpha_i(a_i^k)}^{k+1} = a_i^k) \wedge (a_i^{k+1} = \Delta)], \\ & \forall_{i \in \{1, 2, \dots, m\}} [(a_i^k \neq \Delta) \wedge (z_{\alpha_i(a_i^k)}^k \neq a_i^k) \Rightarrow (a_i^{k+1} = a_i^{k+1})], \\ & \forall_{i \in \{1, 2, \dots, m\}} [(a_i^k = \Delta) \Rightarrow (at_i^q = at_i^k)], \\ & \forall_{i \in \{1, 2, \dots, m\}} [(a_i^k \neq \Delta) \wedge (a_i^q \neq \Delta) \Rightarrow (at_i^q = at_i^k)], \\ & \forall_{i \in \{1, 2, \dots, m\}} [(a_i^k \neq \Delta) \wedge (a_i^q = \Delta) \wedge (at_i^k = \Delta) \Rightarrow (at_i^q = \Delta)], \\ & \forall_{i \in \{1, 2, \dots, m\}} [(a_i^k \neq \Delta) \wedge (a_i^q = \Delta) \wedge (at_i^k \neq \Delta) \wedge (\alpha_i(at_i^k) = \alpha_i(a_i^k)) \Rightarrow (at_i^q = \Delta) \wedge \\ & \quad \wedge (at_{\alpha_i(at_i^k)}^q = at_i^k)], \end{aligned}$$

$$\forall_{i \in \{1,2,\dots,m\}} [(a_i^k \neq \Delta) \wedge (a_i^q = \Delta) \wedge (at_i^k \neq \Delta) \wedge (\alpha_i(at_i^k) \neq \alpha_i(a_i^k))] \Rightarrow (at_i^q = at_i^k),$$

where: m – the number of resources, n – a number of processes,
 $\varphi_i(P_j)$ – the process directly succeeding the process P_j in the i -th priority
 dispatching rule $\sigma_i, \varphi_i(P_j) \in P, \beta_i(P_j)$ – the index of resource directly
 proceeding the resource R_i , in the j -th process route $p_j, \beta_i(P_j) \in \{1,2, \dots, m\},$
 $\alpha_i(P_j)$ – the index of resource directly succeeding the resource R_i , in the j -th
 process route p_j (or multimodal process route mp_j), $\alpha_i(P_j) \in \{1,2, \dots, m\}.$

Note that for a given set of dispatching rules Θ , assuming different initial
 states $S^0 \in SO \subseteq S$, different behaviors can be obtained. So, the quintuple (9) can be
 considered as a model encompassing cyclic steady state behavior shown in Fig. 4.

$$(ST, TO, \Theta, S^0, \delta) \tag{9}$$

The behavior $(ST, TO, \Theta_1, S^0, \delta)$ belongs to $(ST, TO, \Theta_1, S, \delta)$ leading to the cyclic
 steady state shown in Fig. 4 is underlined by a dashed line in Fig. 5. In the considered
 case $(ST, TO, \Theta_1, S, \delta)$, being a part of the state space containing 273 states, one can
 distinguish two cyclic steady states: the first one already mentioned periodicity of which
 is equal to 7 and the second one periodicity of which is equal to 9.

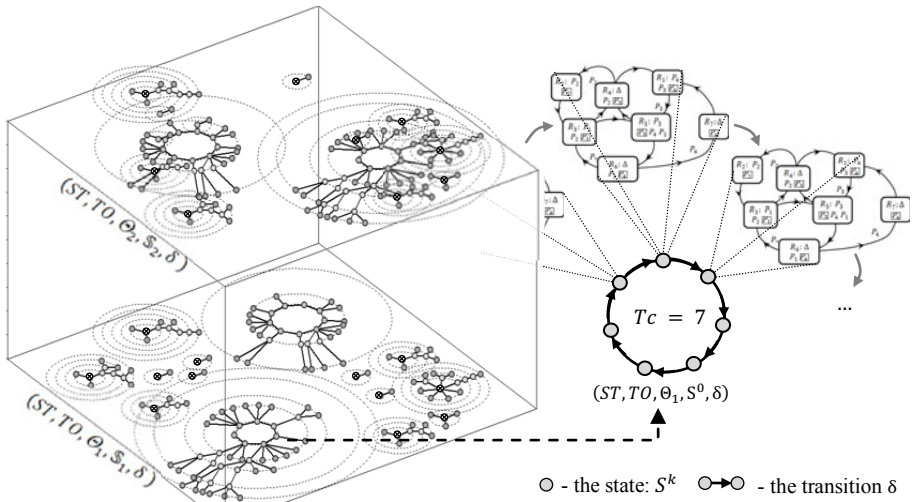


Fig. 5. Illustration of the states space cyclic

Therefore, for different sets of dispatching rules one might consider different cyclic
 steady states, and then different deadlock-free subsets of states. So, a cyclic steady
 state reachability problem can be stated as follows. Consider a set of concurrently
 flowing cyclic processes competing with the access to a set of common shared
 resources. Given a set of pairs (a set of dispatching rules, an initial state) generating
 corresponding sets of deadlock-free states. The question the response we are sought
 is: Does a given cyclic steady state is reachable from assumed other one? Positive

response to this question is shown in Fig. 6. Besides of cyclic steady states one can recognize tree-like structures composed of light color states directly linked or just leading to distinguished by “⊗” deadlock ones.

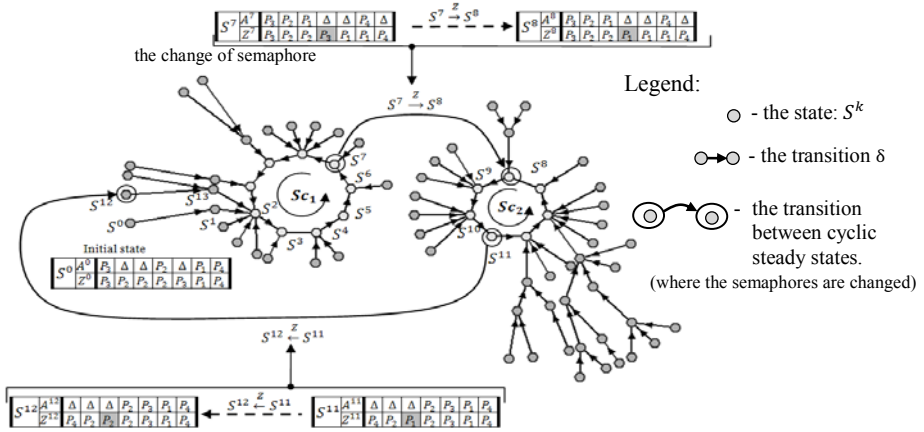


Fig. 6. Illustration of different cyclic steady states linkages

In general case, besides of states determined by a cyclic steady state there are states directly or indirectly leading to the cyclic steady states. Such states leading or being forming a cyclic steady state will be called deadlock-free. Moreover, it is easy to show that a SCCP employing the mutual exclusion protocol as a synchronization mechanism follows the conclusions below.

- “Escapes” from the states belonging to a cyclic steady state are possible only in the case a set of priority rules/semaphores is changed.
- States belonging to an intersection of sets of deadlock-free states lead to different cyclic steady states.

4 Multimodal Processes

Consider the set of cyclic concurrently flowing processes $P = \{P_i \mid i \in \{1, \dots, n\}\}$. The processes interaction is controlled by a mutual exclusion protocol, the instances of which are specified by dispatching rules which determine the way the competitions to the common shared resources are resolved. The considered SCCP is in a cyclic steady state the periodicity of which is equal to Tc .

Let us assume the routings of local processes $p = \{p_i \mid i \in \{1, \dots, n\}\}$ are determined by resource sequences $p_i = (R_{j_1}, R_{j_2}, \dots, R_{j_{lp(i)}})$, $j_k \in \{1, 2, \dots, m\}$ along which the operations are executed. Given are the operation times $t_{i,j} \in \mathbb{N}$ (i.e., the time of process P_i operation on the resource R_j), cycles $Tp_i \in \mathbb{N}$, $i \in \{1, \dots, n\}$ of local cyclic processes as well as values $K_i = Tc/Tp_i$ determining the number of times local processes repeat within the entire SCCP cycle Tc .

For a given SCCP let us consider the following set of multimodal (also sequential) processes $MP = \{mP_i \mid i \in \{1, \dots, nm\}\}$. Multimodal processes interact with each other based on a mutual exclusion protocol while competing to the access to the shared parts of local processes $P_i, i \in \{1, \dots, n\}$. Multimodal processes are determined by routes $mp = \{mp_i \mid i \in \{1, \dots, nm\}\}$ which are sequences of sub sequences (parts) of local cyclic processes $p = \{p_i \mid i \in \{1, \dots, n\}\}$, i.e. $mp_i = (mpr_j(a, b), \dots, mpr_k(c, d))$, where: $mpr_j(a, b) = (crd_a p_j, crd_{a+1} p_j, \dots, crd_b p_j)$, $mpr_k(c, d) = (crd_c p_k, crd_{c+1} p_k, \dots, crd_d p_k)$, $crd_i D = d_i$, for $D = (d_1, d_2, \dots, d_i, \dots, d_w)$.

The questions the responses to which we are looking for can be formulated as follows: What is the minimal period of the cyclic steady state of multimodal processes? Does the period of the cyclic steady state of multimodal processes is less or equal to a given value H ?

In order to illustrate a possible implementation of the above stated framework let us consider the SCCP from Fig. 3 where two different multimodal processes are distinguished by dashed (mP_1) and doted-dashed lines (mP_2). In this case the multimodal processes represent the production routes from Fig. 1. As it was already mentioned, the SCCP can reach two cyclic steady states with periodicity equal to 7 and 9. The periodicity of multimodal processes depends on the current cyclic steady state of the SCCP considered. If its period equals to 9, then the periodicity of multimodal process marked by the dashed line is 9, while the periodicity of multimodal process marked by the dashed-dotted line is equal to 9. In case the SCCP period is equal to 7, then the periodicity of the multimodal process marked by the dashed line is equal to 14, while the periodicity of the multimodal process marked by dashed-dotted line is equal to 7. The question considered is: Is it possible to shorten the periodicity of the multimodal process marked by dashed-dotted (mP_2) line in case the SCCP is in cyclic steady state with its period equal to 9?

The response to this question is positive, see Fig. 7. The multimodal process denoted by the dashed-dotted line can be shortened (from 9 to 7), however, at the cost of the multimodal process marked by dashed line being extended from 9 to 14.

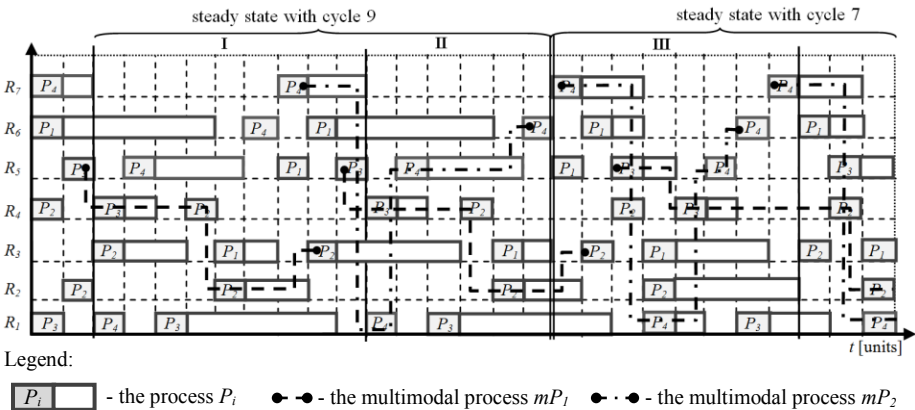


Fig. 7. Solution to cyclic scheduling problem

5 Concluding Remarks

The proposed approach is based on the SCCP concept and involves Diophantine equations modeling. Solutions of the Diophantine equations provide the evaluations of possible cycle periods of the systems under consideration, as well as the evaluation of possible makespans of concurrently executed multimodal processes. Since Diophantine equations can be treated as a set of constraints, descriptive constraint programming languages [1], [2], [4], can be directly implemented. So, besides of a direct the reverse problem formulation can be stated, e.g.: What values of what variables guarantee the SCCP will operate while satisfying the required values of a set of performance indexes specifying the MCP?

Moreover, taking into account non decidability of Diophantine problems, one can easily realize that not all behaviors can be obtained under constraints imposed by the system structure. The similar observation can be made with regard to the system behavior that can be achieved in systems possessing specific structural characteristics.

References

1. Alcaide, D., Chu, C., Kats, V., Levner, E., Sierksma, G.: Cyclic multiple-robot scheduling with time-window constraints using a critical path approach. *European Journal of Operational Research* 177, 147–162 (2007)
2. Bach, I., Bocewicz, G., Banaszak, Z.: Constraint programming approach to time-window and multiresource-constrained projects portfolio prototyping. In: Nguyen, N.T., Borzemski, L., Grzech, A., Ali, M. (eds.) IEA/AIE 2008. LNCS(LNAI), vol. 5027, pp. 767–776. Springer, Heidelberg (2008)
3. Bocewicz, G., Banaszak, Z., Wójcik, R.: Design of admissible schedules for AGV systems with constraints: A logic-algebraic approach. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS(LNAI), vol. 4496, pp. 578–587. Springer, Heidelberg (2007)
4. Birger, R., El-Houssaine, A., Wout, D.: Cyclic scheduling of multiple tours with multiple frequencies for a single vehicle. *International Journal of Logistics Systems and Management* 5(3/4), 214–227 (2009)
5. Cai, X., Li, K.N.: A genetic algorithm for scheduling staff of mixed skills under multi-criteria. *European Journal of Operational Research* 125, 359–369 (2000)
6. Ernst, A.T., Jiang, H.: Krishnamoorthy M., Owens B., Sier D.: An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research* 127, 21–144 (2009)
7. Gaujal, B., Jafari, M., Baykal-Gursoy, M., Alpan, G.: Allocation sequences of two processes sharing a resource. *IEEE Trans. on Robotics and Automation* 11(5), 748–753 (1995)
8. Guy, R.K.: D in Unsolved Problems in Number Theory. In: *Diophantine Equations*, 2nd edn., pp. 139–198. Springer, New York (1994)
9. Pinedo, M.L.: *Planning and scheduling in manufacturing and services*. Springer, New York (2005)
10. Polak, M., Majdzik, P., Banaszak, Z., Wójcik, R.: The performance evaluation tool for automated prototyping of concurrent cyclic processes. *Fundamenta Informaticae* 60(1-4), 269–289 (2004)
11. Sprindzuk, V.G.: Classical Diophantine equations. In: *LNM*, vol. 1559, Springer, Berlin (1993)
12. Smart Nigiel, P.: *The Algorithmic Resolution of Diophantine Equations*. In: *Mathematical Society Student Text*, vol. 41, Cambridge University Press, Cambridge (1998)

Meteorological Phenomena Forecast Using Data Mining Prediction Methods

František Babič¹, Peter Bednár², František Albert¹, Ján Paralič²,
Juraj Bartók³, and Ladislav Hluchý⁴

¹ Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9/B, 042 00 Košice, Slovakia
frantisek.babic@tuke.sk, frantisek.albert@student.tuke.sk

² Centre for Information Technologies, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Nĕmcovej 3, 042 00 Košice, Slovakia
{peter.bednar, jan.paralic}@tuke.sk

³ MicroStep-MIS spol. s.r.o., Čavojského 1, 841 08 Bratislava, Slovakia
jurob@microstep-mis.sk

⁴ Institute of Informatics of the Slovak Academy of Sciences, Dúbravská cesta 9, 845 07 Bratislava, Slovakia
hluchy.ui@savba.sk

Abstract. The occurrence of various meteorological phenomena, such as fog or low cloud cover, has significant impact on many human activities as air or ship transport operations. The management of air traffic at the airports was the main reason to design effective mechanisms for timely prediction of these phenomena. In both these cases meteorologists already use some physical models based on differential equations as simulations. Our goal was to design, implement and evaluate a different approach based on suitable techniques and methods from data mining domain. The selected algorithms were applied on obtained historical data from meteorological observations at several airports in United Arab Emirates and Slovakia. In the first case, the fog occurrence was predicted based on data from METAR messages with algorithms based on neural networks and decision trees. The low cloud cover was forecasted at the national Slovak airport in Bratislava with decision trees. The whole data mining process was managed by CRISP-DM methodology, one of the most accepted in this domain.

Keywords: meteorological data, prediction, decision trees, neural networks.

1 Introduction

The most important meteorological phenomena are clouds, hurricane, lightnings, rain, fog, etc. These events have strong influence on many day-to-day activities, so their effective forecast represents important decision factor for various domains such as traffic and transport, agriculture, tourism and public safety. We have selected management of the air traffic at the specified local airports as our business goal that shall be supported by our solution. The experimental forecast of meteorological phenomena represents very difficult process based on many sources containing raw data in

various formats. These input datasets have to be preprocessed in cooperation with domain experts in order to obtain good-quality data with all necessary attributes and metadata.

The main goal of the presented work is to examine suitability of data mining methods for selected meteorological phenomena forecast in specific conditions of the local airports in United Arab Emirates and Slovakia. Both cases have their specialties and differences that have to be considered during preparation of the data and selection of suitable prediction methods. The obtained results will be compared with already existing and used methods in order to create an effective automatic system for fog and low cloud forecasting at the airports. This Airport Weather System will be deployed by MicroStep-MIS company based on agreed contracts in examined localities.

The whole paper is organized as follows: the first section contains an introduction and brief presentation of data mining domain and relevant Slovak national project called Data Mining Meteo; in the next section we describe several similar or relevant research approaches; the core section is devoted to the detailed description of the whole data mining process in both cases based on CRISP-DM methodology and the paper closes with short summary and a sketch of our future work.

1.1 Data Mining

Data mining covers in our case an application of the whole process with well selected methods to the heterogeneous meteorological data in order to effectively predict specified phenomena [13]. This process can be labeled as knowledge discovery in databases too; this naming is mainly used in academic conditions, but in this paper we understand data mining as the whole knowledge discovery process. This field combines methods from statistics, artificial intelligence, machine learning, database management in various exploitation cases as business (insurance, banking, retail), scientific research (astronomy, medicine), or government security (detection of criminals and terrorists) [14].

CRISP-DM (CRoss Industry Standard Process for Data Mining¹) represents an industry initiative to develop a common and tool-neutral standard for whole data mining process. This methodology is based on collected practical experiences from various companies gained during solving data mining tasks. The whole process can be understood as a life cycle containing six main phases:

- The Business understanding is oriented to specification of business goal, followed with transformation of specified business goal to concrete data mining task(s).
- The Data understanding covers collection of necessary input data for specified tasks; its understanding and initial description.
- The Data preparation contains all performed preprocessing methods, as data transformation, integration, aggregation, reduction, etc.
- In the Modeling phase the selected algorithms are applied on prepared data producing specific models.
- The Evaluation of obtained models based on specific metrics, which depend on the type of used model.

¹ <http://www.crisp-dm.org/>

- The Deployment contains the exploitation of created data mining models in real cases, their adaptation, maintenance and collection of acquired experiences and knowledge.

1.2 DMM Project

The tasks described in this paper have been solved within the Slovak national project called Data Mining Meteo. The project consortium consists of one business partner MicroStep-MIS² with extensive experience in meteorology and two scientific partners with experience in data integration and data mining: Institute of Informatics of the Slovak Academy of Sciences³ and Faculty of Electrical Engineering and Informatics⁴, Technical University of Košice. Each partner has long time experiences in relevant domains as MicroStep-MIS develops, deploys and markets monitoring and information systems in the fields of meteorology, seismology, radiation and emission monitoring and crisis information systems; Institute of Informatics of the Slovak Academy of Sciences is the leading Grid Computing research institution in Slovakia, and has experience in (among others) parallel and distributed computing, grid computing, as well as application of these technologies in the Earth Sciences domain; and the team from the Faculty of Electrical Engineering and Informatics has practical experiences with data-mining in various domains including the mining on unstructured textual data. Additionally, we have experience with the modification of the data-mining algorithms for the distributed GRID environments.

2 State-of-the-Art

The prediction of visibility-reducing fog starts with a common 3D meteorological model executed for a limited region; its outputs are converted using empirical formulae into visibility [8]. This approach by itself cannot achieve results of satisfactory quality and common meteorological models often fail to handle inversion weather conditions, which commonly produce fog. Therefore there are several experimental models in development worldwide, which further process the results of common meteorological model: 1D physical fog modeling methods, statistical post-processing of model outputs [9], [10]. The result is then interpreted by a meteorologist, who takes into account further factors – mainly his/her experience with meteorological situations and local conditions, satellite images, real-time data from meteorological stations suggesting that fog has started to form, or conditions are favorable for the occurrence of fog, conditions of the soil in the target locations, snow cover, recent fog occurrences, etc.

The research group from Italian Aerospace Research Centre developed several fog classifiers based on Bayes networks [1]. The same method was used in [6] for creation of basic network structure that was further adapted to local prediction models. This approach was implemented and tested in the conditions of major Australian

² <http://www.microstep-mis.com/>

³ http://www.ui.savba.sk/index_en.php

⁴ <http://web.tuke.sk/fei-cit/index-a.html>

airports and achieved results represent more than 55 forecasted fogs in row instead of previous operational 7-8 forecasted fog cases. The fog formation and its important parameters were identified based on collected historical dataset from the International Airport of Rio de Janeiro [2].

The Fog forecasting with association rules mining is described in [12]. This paper presents in details the whole process that starts with collection of relevant data; understanding of it; data pre-processing as e.g. feature selection and feature construction, operations with missing values, data transformation; models creation and rules generation. The identified association rules with computed confidence and support represent combination of factors that triggered the fog occurrence. All these rules are further stored in knowledge base and used for relevant expert system creation.

Weather forecasting problem can be stated as the special case of time series prediction. The time series prediction uses algorithm of neural networks able to learn important characteristics from past and present information. The created model is further used for prediction of future states in investigated time series [3], [4]. This approach was used in [5] for fog prediction at the Canberra International Airport. They have created 44-years database of standard meteorological observations and used it to develop, train, and test relevant neural network and to validate obtained results. This neural network was trained to produce forecast for 3h, 6h, 12h and 18h time intervals. Results with cross-validated mean value 0.937 in 3 lead times indicate good forecasting ability of used neural network that was robust to error perturbations in the meteorological data.

Y. Radhika and M. Shashi in [7] proposed an exploitation of Support Vector Machine method for weather prediction. The time series datasets of daily maximum temperature in a location were analyzed to predict the maximum of the next day. The performance of Support Vector Machine was compared with Multi Layer Perceptron using the Mean Square Error metrics. In the first case the error was in the range of 7,07 to 7,56, whereas the second one varies between 8,07 and 10,2.

The delays in aircraft traffic caused by weather conditions were predicted at the Frankfurt airport within algorithms of neural networks, decision trees, linear regression and fuzzy clustering [11]. In this case value of travel time was used as target value for the algorithms listed above. The obtained results documented easier interpretation of decision trees and clustering results; as well as up to 20% higher prediction accuracy as with simple mean estimators.

3 Proposed Approach for Detection of Fog and Low Cloud Cover

The performed data mining process was divided into two branches based on two specified business goals - fog forecast and low cloud cover prediction. Several operations were very similar, but each direction has its specifications and dependencies that have to be solved.

3.1 Business Understanding

In the first case we focused on the best prediction of fog occurrence at given locality, where any suitable historical data can be used for it. We identified two possible alternatives with different costs, i.e. more costly to predict a fog which does not appear than do not predict a fog, which suddenly appears. We have specified a binary classification: fog or no fog, as our primary data mining task. Even if our main goal was to get a prediction of the best possible quality, also the interpretation of the rules used for prediction can be interesting - the ability to comprehensively describe the processes leading to occurrence of fog.

The prediction of low cloud cover is quite different; we have defined five classification categories that each represents relevant fragment of the sky covered by low cloud. The sky is divided into eight fragments, so e.g. low cloud cover index is 2/8 and it means that two sky fragments (out of 8) are covered by low cloud.

3.2 Data Understanding

Data understanding phase started with the selection of data relevant for the specified problems. We have investigated several available data sources as sets of physical quantities measured automatically by meteorological stations or radars; sets of physical quantities computed by standard physical models, etc.

When we had the identified data available, we performed an initial data examination in order to verify the quality of the data. These operations were extended with a calculation of the basic statistics for key attributes and their correlations.

We have selected different input samples to test and evaluate suitability of relevant data mining methods for our goals. For the purposes of fog forecasting we collected historical data geographically covering the area of 10 airports in United Arab Emirates mainly located around Dubai and north coastline with time span and granularity of 10 years measured each one-hour. The quality of available meteorological data was low with high number of missing records (in average 30% of records per airport, for some airports as much as 90%).

In the second case the historical dataset contained data from selected ceilometers and relevant METAR messages describing weather conditions at the Bratislava airport in Slovakia for three past years (2007 - 2010). The ceilometers are sensors routinely deployed at the airports for measurement of cloud base heights above the points of their installation. These three ceilometers determine the height of a cloud base through laser in 15s intervals. Sometimes the ceilometers data is used to determine also the cloud amount using the FAA method (approach developed by U.S. Federal Aviation Administration that uses the widely recognized Rational Coefficient to describe cloud cover), where the result is a simple combination of laser reflection counts in different height categories.

3.3 Data Preparation

Data pre-processing is usually the most complex and also most time consuming phase of the whole data mining process; usually taking 60 to 70 percent of the overall time. We applied necessary pre-processing operations to obtain ready datasets for

implementation of selected mining methods. Meteorological data from all sources (i.e. data extracted from messages, meteorological stations and physical model predictions) were integrated into one relational database. The performed operations were very similar for both cases with little modifications based on characteristics of the data and related data mining tasks.

The first step was data extraction from the meteorological messages broadcasted from the meteorological stations in METAR format. This format of the messages was fixed with standard codes denoting the parts of the messages and data values. The output of this task was structured data stored in a relational database with raw extracted data. Each record in the integrated database has assigned valid from/to time interval and 3D coordinates of measured area (i.e. ranges for longitude, latitude and altitude). Since each data source had different data precision and/or granularity, the goal of this operation was to interpolate measured values and compute additional data for the requested area and time with the specified data granularity. The same approach was used for replacement of missing values. We have selected a representative sample for next modeling phase. Reduction was necessary by reason of the technical restrictions inherent to some methods, but it can also lead to simplification of the task at hand by removing irrelevant attributes and records, thus even increasing the quality of the results.

The consultations with the domain experts resulted into specified valid ranges of values and detected invalid data. Out-of-range data were considered as missing values. Also we identified the principal problem specific for fog prediction - unbalanced distribution of fog-positive and fog-negative examples; fog occurred only in 0.36% of cases. We have tried to integrate additional data source from Climatological Database System (CLDB⁵) but still data quality had to be improved.

We identified key attributes for both cases:

- The fog prediction e.g. actual weather at the airport as fog, drizzle, rain; cloud amount in first layer; overall cloud amount; visibility; relative humidity; etc. – parameters from METAR messages
- The low cloud cover prediction e.g. detection status, CAVOK (Ceiling And Visibility Okay indicates no cloud below 1 500m, visibility of 10km and no cumulonimbus at any level), detection level 1, detection level 2, detection level 3 – parameters from ceilometers and METAR too.

Additionally, data were enhanced with some derived attributes computed using empiric formulas, as a ratio of physical attributes or trend. These attributes included information about the fog situation in neighboring airports (average for 3 or 5 closest airports to the target area) and relative humidity computed empirically from temperature and dew point. Trend attributes were computed for temperature, dew point, relative humidity, difference between temperature and dew point and pressure.

In the second case of low cloud cover we have integrated data from ceilometers and selected METAR attributes into one dataset for modeling phase. This aggregation was realized based on assignment of relevant METAR message (every half an hour) to each ceilometer record (every 15s). The final dataset contains 1081 columns: 120 time points x 3 ceilometers x 3 attributes for each ceilometer + 1 target attribute CTOT extracted from METAR messages.

⁵ <http://www.microstep-mis.com/index.php?lang=en&site=src/products/meteorology/cldb>

In both cases we started with initial time series of five records that were further modified in several iterations based on modeling phase results.

All these pre-processing operations were realized within SQL database, IBM SPSS software and own designed and implemented applications in Microsoft .Net or C#.

3.4 Modeling

Modeling phase represents the core of the whole data mining process when selected data mining methods are applied on pre-processed data. Our models are simple predictors for time series, where the prediction of outputs for time $t+1, \dots, t+K$ is based on the sequence of historical data (i.e. time “window”) from time $\dots, t-2, t-1, t$. Prediction of outputs is limited to future one hour, i.e. $K = 1$.

There is a whole range of prediction methods – from statistical methods to artificial intelligence methods, like linear or logistical regression models, Support Vector Machine, neural nets, probabilistic models, decision/regression trees and lists, etc. We have tested several of these methods provided in the IBM SPSS data mining environment. Finally we selected decision trees models and neural networks. In order to obtain optimal results, all parameters of these algorithms were tuned by testing several strategies to divide input dataset into training and test set. For example, this division was important for fog prediction by reason of the unbalanced distribution for fog-positive and fog-negative cases. We have tested three types of distribution: random division with 90% examples for training and 10% for testing; the same random division with stratification and 10-fold cross validation with stratification.

In the case of low cloud cover we have realized already the initial experiments with decision trees algorithms as C5.0 or CHAID based on distribution into 50% for training and 50 % for testing.

3.5 Evaluation

The created models for fog forecasting were evaluated using these measures:

- Recall = $TP / (TP + FN)$;
 - False alarm = $FP / (TP + FP)$;
 - True skill score = recall – false alarm;
- TP (FP) is the number of true (false) positive
 TN (FN) is the number of true (false) negative examples respectively.

Table 1. Accuracy of generated models (90% training set, 10% testing set) for fog prediction

Model ()	Recall	False Alarm	True skill score
Decision tress	0.77 ± 0.8	0.44 ± 0.14	0.33 ± 0.19
Neural networks	0.68 ± 0.8	0.41 ± 0.1	0.26 ± 0.12

The results presented in Table 1 represent prediction of continuous fog occurrence, i.e. if fog = 1 in t_i-1 (previous record in the timeframe) then fog = 1 in t_i (target attribute). In the next step we eliminated just these records and we continued with experiments on new cleansed dataset that resulted into models with lower prediction quality, but covered more representative and required situations for the prediction at the

airports. We have continually experimented with specified data distribution in order to balance positive and negative records of fog occurrence. In current training dataset there were only 0.2% of positive cases of fog. We have tried to balance data using simple re-sampling or 10-fold cross validation with stratification (True skill score = -41.44 ± 0.049).

The obtained results are plausible and comparable with the other existing methods, but they still need improvement in several directions, e.g. quality of input data can be improved with inclusion of data extracted from satellite images; utilization of clustering analyses for identification of representative patterns from all negative records of fog occurrence in the same quantity as positive records; or understanding of created models for domain experts.

In the case of low cloud cover we have started basic experiments within decision trees algorithm (C5.0) and 10-fold cross validation for evaluation of generated models. Based on first results, we have implemented stratified 10-fold cross validation in order to balance training data for each target value. The generated models had accuracy around 80%, but we identified several inconsistencies in data characteristics, see Table 2.

Table 2. Coincidence matrix for C5.0 model

CTOT value (original vs. predicted)	-1	0.0	1.5	3.5	6.0	8.0	9.0
-1	2016	77	128	128	186	14	0
0.0	1	14	0	1	5	2	0
1.5	4	0	8	1	1	0	0
3.5	11	2	8	258	29	4	0
6.0	47	5	5	51	768	70	0
8.0	4	0	0	1	47	156	0
9.0	0	0	0	0	0	0	1

These results are plausible too, but you can see the problem with relatively high number of false classified records, mainly in category -1. This fact can be caused by unbalance distribution of target attribute; high number of missing values in target attribute that are labeled with -1 value; low number of records for category labeled with 9 – other meteorological phenomenon, no low cloud. Based on these findings, we have realized several other experiments when we eliminated records without target value or we joined three target categories -1, 0.0 and 9.0 into one category. The main problem was relatively high number of missing values in the target attribute CTOT that will be solved as one part of our future work.

3.6 Deployment

The main aim of this last phase is to design and develop a practical deployment plan for the best generated models. This plan covers the strategy for implementation, monitoring and maintenance in order to provide effective application. In our case, the generated models with detailed description will be used as integrated part of Airport Weather System developed in MicroStep-MIS company specialized in design, development and manufacturing of various monitoring and information systems. The whole data mining process will be evaluated in order to identify “best practices” for future projects of similar character.

4 Conclusion

The prediction of various meteorological phenomena represents important factor for many human activities. We have predicted fog occurrence at several airports in United Arab Emirates and low cloud cover at the Slovak national airport in Bratislava. In both cases we have collected necessary historical data, mainly from METAR messages and ceilometers. All data were preprocessed and verified based on selected data mining methods: decision trees and neural networks.

We used CRISP-DM methodology for the realization of the whole data mining process as one of the most used methodology in this domain. This approach contains six main phases with relevant operations, conditions and opportunities; see section 3 for detailed information. We have implemented the whole chain of data pre-processing operations which extracts and integrates data from various meteorological sources as our own application that will be presented as one of the core outputs of Data Mining Meteo project. The description of used methods and their parameters can be used in the future as helping or inspiring materials for someone that will perform similar experiments and based on available information it will be possible to prevent the same inappropriate steps to save the money, time and energy. According to preliminary results presented in section 3.5, our models can be compared with the other existing methods based on the global physical model and empirical rules.

The future plans contain several interesting and perspective tasks mainly oriented in data processing domain as collection and integration of additional data sources (e.g. satellite images); experimental evaluation of various balance strategies (e.g. clustering analyses resulted into representative negative patterns) for positive and negative fog records; descriptive data mining; and the additional experiments with additional algorithms for low cloud cover.

Acknowledgments. The work presented in the paper was supported by the Slovak Research and Development Agency under the contract No. VMSP-P-0048-09 (40%); the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic under grant No. 1/0042/10 (30%) and project implementation: Development of the Center of Information and Communication Technologies for Knowledge Systems (ITMS project code: 26220120030) (30%) supported by the Research & Development Operational Program funded by the ERDF.

References

1. Zazzaro, G., Pisano, F.M., Mercogliano, P.: Data Mining to Classify Fog Events by Applying Cost-Sensitive Classifier. In: International Conference on Complex, Intelligent and Software Intensive Systems 2010, pp. 1093–1098 (2010) ISBN 978-1-4244-5917-9
2. Ebecken, F.F.: Fog Formation Prediction In Coastal Regions Using Data Mining Techniques. In: International Conference On Environmental Coastal Regions, Cancun, Mexico, vol (2), pp. 165–174 (1998) ISBN 1-85312-527-X
3. Acosta, G., Tosini, M.: A Firmware Digital Neural Network for Climate Prediction Applications. In: Proceedings of IEEE International Symposium on Intelligent Control 2001, Mexico City, Mexico (2001) ISBN 0-7803-6722-7

4. Koskela, T., Lehtokangas, M., Saarinen, J., Kaski, K.: Time Series Prediction With Multi-layer Perceptron, FIR and Elman Neural Networks. In: Proceedings of the World Congress on Neural Networks, pp. 491–496. INNS Press, San Diego (1996)
5. Fabbian, D., de Dear, R., Lelleyett, S.: Application of Artificial Neural Network Forecasts to Predict Fog at Canberra International Airport. *Weather and Forecasting* 22(2), 372–381 (2007)
6. Weymouth, G.T., et al.: Dealing with uncertainty in fog forecasting for major airports in Australia. In: 4th Conference on Fog, Fog Collection and Dew, La Serena, Chile, pp. 73–76 (2007)
7. Radhika, Y., Shashi, M.: Atmospheric Temperature Prediction Using SVM. *International Journal of Computer Theory and Engineering* 1(1), 1793–8201 (2009)
8. Gulpepe, I., Müller, M.D., Boybeyi, Z.: A new visibility parameterization for warm fog applications in numerical weather prediction models. *J. Appl. Meteor.* 45, 1469–1480 (2006)
9. Bott, A., Trautmann, T.: PAFOG - a new efficient forecast model of radiation fog and low-level stratiform clouds. *Atmos. Research* 64, 191–203 (2002)
10. COST 722 - Short range forecasting methods of fog, visibility and low clouds. Final Report, COST Office, Brussels, Belgium (2007)
11. Rehm, F.: Prediction of Aircraft Delay at Frankfurt Airport as a Function of Weather. Presentation from German Aerospace Center, Germany (2004)
12. Viademonte, S., Burstein, F., Dahni, R., Williams, S.: Discovering Knowledge from Meteorological Databases: A Meteorological Aviation Forecast Study. In: Kambayashi, Y., Winiwarter, W., Arikawa, M. (eds.) *DaWaK 2001*. LNCS, vol. 2114, pp. 61–70. Springer, Heidelberg (2001)
13. Hluchý, L., Habala, O., Tran, D.V., Ciglan, M.: Hydro-meteorological scenarios using advanced data mining and integration. In: *The Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 7, pp. 260–264. IEEE Computer Society, Los Alamitos (2009) ISBN 978-0-7695-3735-1
14. Clifton, C.: *Encyclopedia Britannica: Definition of Data Mining* (2010), <http://www.britannica.com>

Artificial Immune Clustering Algorithm to Forecasting Seasonal Time Series

Grzegorz Dudek

Faculty of Electrical Engineering, Czestochowa University of Technology,
Al. Armii Krajowej 17, 42-200 Czestochowa, Poland
dudek@el.pcz.czest.pl

Abstract. This paper concentrates on the forecasting time series with multiple seasonal periods using new immune inspired method. Proposed model includes two populations of immune memory cells – antibodies, which recognize patterns of the time series sequences represented by antigens. The empirical probabilities, that the pattern of forecasted sequence is detected by the j th antibody from the first population while the corresponding pattern of input sequence is detected by the i th antibody from the second population, are computed and applied to the forecast construction. The suitability of the proposed approach is illustrated through an application to electrical load forecasting.

Keywords: artificial immune system, cluster analysis, seasonal time series forecasting, similarity-based methods.

1 Introduction

In general, a time series can be thought of as consisting of four different components: trend, seasonal variations, cyclical variations, and irregular component. The specific functional relationship between these components can assume different forms. Usually they combine in an additive or a multiplicative fashion.

Seasonality is defined to be the tendency of time series data to exhibit behavior that repeats itself every m periods. The difference between a cyclical and a seasonal component is that the latter occurs at regular (seasonal) intervals, while cyclical factors have usually a longer duration that varies from cycle to cycle. Seasonal patterns of time series can be examined via correlograms or periodograms based on a Fourier decomposition.

Many economical, business and industrial time series exhibit seasonal behavior. A variety of methods have been proposed for forecasting seasonal time series. These include: exponential smoothing, seasonal ARMA, artificial neural networks, dynamic harmonic regression, vector autoregression, random effect models, and many others.

The proposed approach belongs to the class of similarity-based methods [1] and is dedicated to forecasting time series with multiple seasonal periods. The forecast here is constructed using analogies between sequences of the time series with periodicities. An artificial immune system (AIS) is used to detection of similar patterns of

sequences. The clusters of patterns are represented by antibodies (AB). Two population of ABs are created which recognize two populations of patterns (antigens) – input ones and forecasted ones. The empirical probabilities, that the pattern of forecasted sequence is detected by the j th AB from the first population while the corresponding pattern of input sequence is detected by the i th AB from the second population are computed and applied to the forecast construction. Another AIS to forecasting of seasonal time series was presented in [2].

The merits of AIS lie in its pattern recognition and memorization capabilities. The application areas for AIS can be summarized as [3]: learning (clustering, classification, recognition, robotic and control applications), anomaly detection (fault detection, computer and network security applications), and optimization (continuous and combinatorial). Antigen recognition, self-organizing memory, immune response shaping, learning from examples, and generalization capability are valuable properties of immune systems which can be brought to potential forecasting models. A good introduction to AIS are [4] and [5].

2 Similarity-Based Forecasting Methods

The similarity-based (SB) methods use analogies between sequences of the time series with periodicities. A course of a time series can be deduced from the behavior of this time series in similar conditions in the past or from the behavior of other time series with similar changes in time. In the first stage of this approach, the time series is divided into sequences of length n , which usually contain one period. Fig. 1 shows the periodical time series, where we can observe yearly, weekly and daily variations. This series represents hourly electrical loads of the Polish power system. Our task is to forecast the time series elements in the daily period, so the sequences include 24 successive elements of daily periods.

In order to eliminate trend and seasonal variations of periods longer than n (weekly and annual variations in our example), the sequence elements are preprocessed to obtain their patterns. The pattern is a vector with components that are functions of real time series elements. The input and output (forecast) patterns are defined: $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$ and $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_n]^T$, respectively. The patterns are paired $(\mathbf{x}_i, \mathbf{y}_i)$, where \mathbf{y}_i is a pattern of the time series sequence succeeding the sequence represented by \mathbf{x}_i and the interval between these sequences (forecast horizon τ) is constant. The SB methods are based on the following assumption: if the process pattern \mathbf{x}_a in a period preceding the forecast moment is similar to the pattern \mathbf{x}_b from the history of this process, then the forecast pattern \mathbf{y}_a is similar to the forecast pattern \mathbf{y}_b .

Patterns \mathbf{x}_a , \mathbf{x}_b and \mathbf{y}_b are determined from the history of the process. Pairs \mathbf{x}_a – \mathbf{x}_b and \mathbf{y}_a – \mathbf{y}_b are defined in the same way and are shifted in time by the same number of series elements. The similarity measures are based on the distance or correlation measures.

The way of how the \mathbf{x} and \mathbf{y} patterns are defined depends on the time series nature (seasonal variations, trend), the forecast period and the forecast horizon. Functions transforming series elements into patterns should be defined so that patterns carry most information about the process. Moreover, functions transforming forecast sequences into patterns \mathbf{y} should ensure possibility of calculation of real forecast of time series elements.

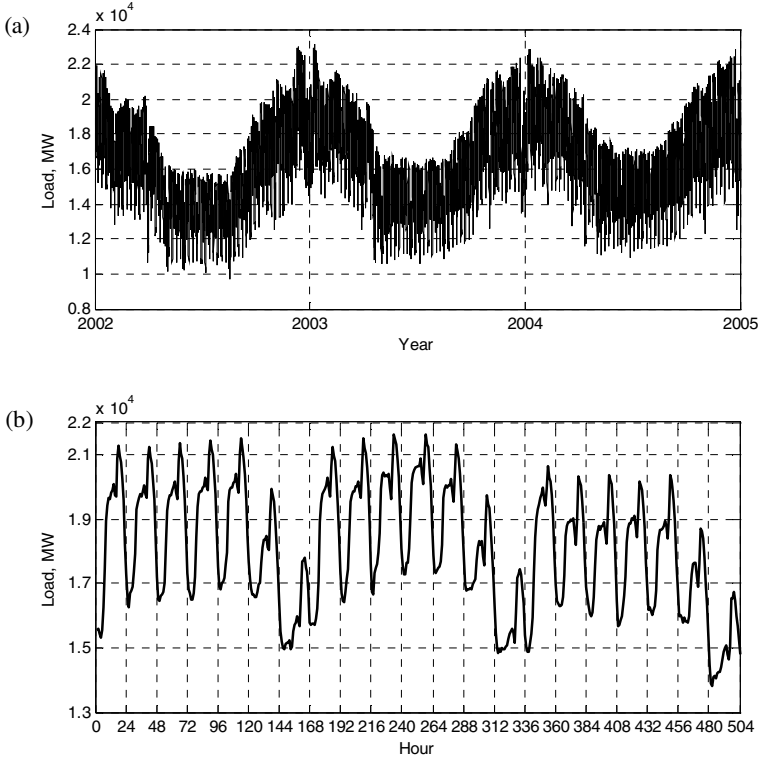


Fig. 1. The load time series of Polish power system in three year (a) and three week (b) intervals

The forecast pattern $\mathbf{y}_i = [y_{i,1} \ y_{i,2} \ \dots \ y_{i,n}]$ encodes the following real time series elements z in the forecast period $i + \tau$. $\mathbf{z}_{i+\tau} = [z_{i+\tau,1} \ z_{i+\tau,2} \ \dots \ z_{i+\tau,n}]$, and the input pattern $\mathbf{x}_i = [x_{i,1} \ x_{i,2} \ \dots \ x_{i,n}]$ maps the time series elements in the period i preceding the forecast period: $\mathbf{z}_i = [z_{i,1} \ z_{i,2} \ \dots \ z_{i,n}]$. In general, the input pattern can be defined on the basis of a sequence longer than one period, and the time series elements contained in this sequence can be selected in order to ensure the best quality of the model. Vectors \mathbf{y} are encoded using actual process parameters Ψ_i (from the nearest past), which allows to take into consideration current variability of the process and ensures possibility of decoding.

Some functions mapping the original feature space Z into the pattern spaces X and $Y - f_x : Z \rightarrow X$ and $f_y : Z \rightarrow Y$ – are presented below.

$$f_x(z_{i,t}, \Psi_i) = \frac{z_{i,t} - \bar{z}_i}{\sqrt{\sum_{l=1}^n (z_{i,l} - \bar{z}_i)^2}}, \quad f_y(z_{i,t}, \Psi_i) = \frac{z_{i+\tau,t} - \bar{z}_i}{\sqrt{\sum_{l=1}^n (z_{i+\tau,l} - \bar{z}_i)^2}} \quad (1)$$

$$f_x(z_{i,t}, \Psi_i) = \frac{z_{i,t}}{z'}, \quad f_y(z_{i,t}, \Psi_i) = \frac{z_{i+\tau,t}}{z''} \quad (2)$$

$$f_x(z_{i,t}, \Psi_i) = z_{i,t} - z', \quad f_y(z_{i,t}, \Psi_i) = z_{i+\tau,t} - z'' \quad (3)$$

where: $i = 1, 2, \dots, M$ – the period number, $t = 1, 2, \dots, n$ – the time series element number in the period i , τ – the forecast horizon, $z_{i,t}$ – the t th time series element in the period i , \bar{z}_i – the mean value of elements in period i , $z' \in \{\bar{z}_i, z_{i-1,t}, z_{i-7,t}, z_{i,t-1}\}$, $z'' \in \{\bar{z}_i, z_{i+\tau-1,t}, z_{i+\tau-7,t}\}$, Ψ_i – the set of coding parameters such as \bar{z}_i , z' and z'' .

The function f_x defined using (1) expresses normalization of the vectors \mathbf{z}_i . After normalization they have the unity length, zero mean and the same variance. When we use the standard deviation of the vector \mathbf{z}_i components in the denominator of equation (1), we receive vector \mathbf{x}_i with the unity variance and zero mean.

The components of the x-patterns defined using equations (2) and (3) express, respectively, indices and differences of time series elements in the following whiles of the i th period.

Forecast patterns are defined using analogous functions to input pattern functions f_x , but they are encoded using the time series elements determined from the process history, what enables decoding of the forecasted vector $\mathbf{z}_{i+\tau}$ after the forecast of pattern \mathbf{y} is determined. To calculate the time series element values on the basis of their patterns we use the inverse functions: $f_x^{-1}(x_{i,t}, \Psi_i)$ or $f_y^{-1}(y_{i,t}, \Psi_i)$.

If for a given time series the statistical analysis confirms the hypothesis that the dependence between similarities of input patterns and similarities between forecast patterns paired with them, are not caused by random character of the sample, it justifies the sense of building and using models based on the similarities of patterns of this time series. The statistical analysis of pattern similarities is described in [1].

The forecasting procedure in the case of SB methods can be summarized as follows: (i) elimination of the trend and seasonal variations of periods longer than n using pattern functions f_x and f_y , (ii) forecasting the pattern \mathbf{y} using similarities between x-patterns, and (iii) reconstruction the time series elements from the forecasted pattern \mathbf{y} using the inverse function f_y^{-1} .

3 Immune Inspired Forecasting Model

The proposed AIS contains immune memory consisting of two populations of ABs. The population of x-antibodies (ABx) detects antigens representing patterns $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$: AGx, while the population of y-antibodies (ABy) detects antigens representing patterns $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$: AGy. The vectors \mathbf{x} and \mathbf{y} form the epitopes of AGs and paratopes of ABs. ABx has the cross-reactivity threshold r defining the AB recognition region. This recognition region is represented by the n -dimensional hypersphere of radius r with center at the point \mathbf{x} . Similarly ABy has the recognition region of radius s with center at the point \mathbf{y} . The cross-reactivity thresholds are adjusted individually during training. The recognition regions contain AGs with similar epitopes.

AG can be bound to many different ABs of the same type (x or y). The strength of binding (affinity) is dependent on the distance between an epitope and a paratope. AB represents a cluster of similar AGs in the pattern space X or Y . The clusters are overlapped and their sizes depend on the similarity between AGs belonging to them, measured in the both pattern spaces X and Y . The k th ABx can be written as a pair $\{\mathbf{p}_k, r_k\}$, where $\mathbf{p}_k = \mathbf{x}_k$, and the k th ABY as $\{\mathbf{q}_k, s_k\}$, where $\mathbf{q}_k = \mathbf{y}_k$.

After the two population of immune memory have been created, the empirical conditional probabilities $P(AB_{y_k} | AB_{x_j}), j, k = 1, 2, \dots, N$, that the i th AGy stimulates (is recognized by) the k th ABY, when the corresponding i th AGx stimulates the j th ABx, are determined. These probabilities are calculated for each pair of ABs on the basis of recognition of the training population of AGs.

In the forecasting phase the new AGx, representing pattern \mathbf{x}^* , is presented to the trained immune memory. The forecasted pattern \mathbf{y} paired with \mathbf{x}^* is calculated as the mean of PCy paratopes weighted by the conditional probabilities and affinities.

The detailed algorithm of the immune system to forecasting seasonal time series is described below.

Step 1. Loading of the training population of antigens. An AGx represents a single \mathbf{x} pattern, and AGy represents a single \mathbf{y} pattern. Both populations AGx and AGy are divided into training and test parts in the same way. Immune memory is trained using the training populations, and after learning the model is tested using the test populations.

Step 2. Generation of the antibody population. The AB populations are created by copying the training populations of AGs (ABs and AGs have the same structure). Thus the paratopes take the form: $\mathbf{p}_k = \mathbf{x}_k, \mathbf{q}_k = \mathbf{y}_k, k = 1, 2, \dots, N$. The number of AGs and ABs of both types is the same as the number of learning patterns.

Step 3. Calculation of the cross-reactivity thresholds of x-antibodies. The recognition region of k th ABx should be as large as possible and cover only the AGx that satisfy two conditions: (i) their epitops \mathbf{x} are similar to the paratope \mathbf{p}_k , and (ii) the AGy paired with them have epitopes \mathbf{y} similar to the k th ABY paratope – \mathbf{q}_k .

The measure of similarity of the i th AGx to the k th ABx is the affinity:

$$a(\mathbf{p}_k, \mathbf{x}_i) = \begin{cases} 0, & \text{if } d(\mathbf{p}_k, \mathbf{x}_i) > r_k \text{ or } r_k = 0 \\ 1 - \frac{d(\mathbf{p}_k, \mathbf{x}_i)}{r_k}, & \text{otherwise} \end{cases}, \tag{4}$$

where: $d(\mathbf{p}_k, \mathbf{x}_i)$ is the distance between vectors \mathbf{p}_k and \mathbf{x}_i , $a(\mathbf{p}_k, \mathbf{x}_i) \in [0, 1]$. $a(\mathbf{p}_k, \mathbf{x}_i)$ informs about the degree of membership of the i th AGx to the cluster represented by the k th ABx.

The similarity of the i th AGy to the k th ABY mentioned in (ii) is measured using the forecast error of the time series elements encoded in the paratope of the k th ABY. These elements are forecasted using the epitope of the i th AGy:

$$\delta_{k,i} = \frac{100}{n} \sum_{t=1}^n \frac{|z_{k+\tau,t} - f_y^{-1}(y_{i,t}, \Psi_k)|}{z_{k+\tau,t}}, \quad (5)$$

where: $z_{k+\tau,t}$ – the t th time series element of the period $k+\tau$ which is encoded in the paratope of the k th ABY: $q_{k,t} = f_y(z_{k+\tau,t}, \Psi_k)$, $f_y^{-1}(y_{i,t}, \Psi_k)$ – the inverse function of pattern \mathbf{y} returning the forecast of the time series element $z_{k+\tau,t}$ using the epitope of the i th AGy.

If the condition $\delta_{k,i} \leq \delta_y$ is satisfied, where δ_y is the error threshold value, it is assumed that the i th AGy is similar to the k th ABY, and i th AGx, paired with this AGy, is classified to class 1. When the above condition is not met the i th AGx is classified to class 2. Thus class 1 indicates the high similarity between ABY and AGY. The classification procedure is performed for each ABX.

The cross-reactivity threshold of the k th ABx is defined as follows:

$$r_k = d(\mathbf{p}_k, \mathbf{x}_A) + c[d(\mathbf{p}_k, \mathbf{x}_B) - d(\mathbf{p}_k, \mathbf{x}_A)], \quad (6)$$

where B denotes the nearest AGx of class 2 to the k th ABx, and A denotes the furthest AGx of class 1 satisfying the condition $d(\mathbf{p}_k, \mathbf{x}_A) < d(\mathbf{p}_k, \mathbf{x}_B)$. The parameter $c \in [0, 1]$ allows to adjust the cross-reactivity threshold value from $r_{k\min} = d(\mathbf{p}_k, \mathbf{x}_A)$ to $r_{k\max} = d(\mathbf{p}_k, \mathbf{x}_B)$.

Step 4. Calculation of the cross-reactivity thresholds of y-antibodies. The cross-reactivity threshold of the k th ABY is calculated similarly to the above:

$$s_k = d(\mathbf{q}_k, \mathbf{y}_A) + b[d(\mathbf{q}_k, \mathbf{y}_B) - d(\mathbf{q}_k, \mathbf{y}_A)], \quad (7)$$

where B denotes the nearest AGy of class 2 to the k th ABY, and A denotes the furthest AGy of class 1 satisfying the condition $d(\mathbf{q}_k, \mathbf{y}_A) < d(\mathbf{q}_k, \mathbf{y}_B)$. The parameter $b \in [0, 1]$ plays the same role as the parameter c .

The i th AGy is classified to class 1, if for the i th AGx paired with it, there is $\varepsilon_{k,i} \leq \varepsilon_x$, where ε_x is the threshold value and $\varepsilon_{k,i}$ is the forecast error of the time series elements encoded in the paratope of the k th ABx. These elements are forecasted using the epitope of the i th AGx:

$$\varepsilon_{k,i} = \frac{100}{n} \sum_{t=1}^n \frac{|z_{k,t} - f_x^{-1}(x_{i,t}, \Psi_k)|}{z_{k,t}}, \quad (8)$$

where: $z_{k,t}$ – the t th time series element of the period k which is encoded in the paratope of the k th ABx: $p_{k,t} = f_x(z_{k,t}, \Psi_k)$, $f_x^{-1}(x_{i,t}, \Psi_k)$ – the inverse function of pattern \mathbf{x} returning the forecast of the time series element $z_{k,t}$ using the epitope of the i th AGx.

The i th AGy is recognized by the k th ABY if affinity $a(\mathbf{q}_k, \mathbf{y}_i) > 0$, where:

$$a(\mathbf{q}_k, \mathbf{y}_i) = \begin{cases} 0, & \text{if } d(\mathbf{q}_k, \mathbf{y}_i) > s_k \text{ or } s_k = 0 \\ 1 - \frac{d(\mathbf{q}_k, \mathbf{y}_i)}{s_k}, & \text{otherwise} \end{cases} \quad (9)$$

$a(\mathbf{q}_k, \mathbf{y}_i) \in [0, 1]$ expresses the degree of membership of pattern \mathbf{y}_i to the cluster represented by the k th ABY.

Procedure for determining the threshold s_k is thus analogous to the procedure for determining the threshold r_k . The recognition region of k th ABY is as large as possible and covers AGy that satisfy two conditions: (i) their epitops \mathbf{y} are similar to the paratope \mathbf{q}_k , and (ii) the AGx paired with them have epitops \mathbf{x} similar to the k th ABx paratope \mathbf{p}_k .

This way of forming clusters in pattern space X (Y) makes that their sizes are dependent on the dispersion of y -patterns (x -patterns) paired with patterns belonging to these clusters. Another pattern \mathbf{x}_i (\mathbf{y}_i) is appended to the cluster ABx_k (ABy_k) (this is achieved by increasing the cross-reactivity threshold of AB representing this cluster), if the pattern paired with \mathbf{x}_i (\mathbf{y}_i) is sufficiently similar to the paratope of the k th ABy_k (ABx_k). The pattern is considered sufficiently similar to the paratope, if it allows to forecast the paratope with an error no greater than the threshold value. This ensures that the forecast error for the pattern \mathbf{x} (\mathbf{y}) has a value not greater than ε_x (δ_y). Lower error thresholds imply smaller clusters, lower bias and greater variance of the model.

MAPE here is used as an error measure ((5) and (8)) but other error measures can be used.

Step 5. Calculation of the empirical conditional probabilities $P(ABy_k|ABx_j)$. After the clustering of both spaces is ready, the successive pairs of antigens (AGx_i, AGy_i), $i = 1, 2, \dots, N$, are presented to the trained immune memory. The stimulated ABx and ABY are counted and the empirical frequencies of ABy_k given ABx_j , estimating conditional probabilities $P(ABy_k|ABx_j)$, are determined.

Step 6. Forecast procedure. In the forecast procedure a new AGx, representing the pattern \mathbf{x}^* , is presented to the immune memory. Let Ω be a set of ABx stimulated by this AGx. The forecasted pattern \mathbf{y} corresponding to \mathbf{x}^* is estimated as follows:

$$\hat{\mathbf{y}} = \sum_{k=1}^N w_k \mathbf{q}_k, \quad (10)$$

where

$$w_k = \frac{\sum_{j \in \Omega} P(ABy_k | ABx_j) a(\mathbf{p}_j, \mathbf{x}^*)}{\sum_{l=1}^N \sum_{j \in \Omega} P(ABy_l | ABx_j) a(\mathbf{p}_j, \mathbf{x}^*)}. \quad (11)$$

The forecast is calculated as the weighted mean of paratopes \mathbf{q}_k . Weights w express the products of the affinity of stimulated memory cells ABx to the AGx and probabilities $P(ABy_k | ABx_j)$. $\sum_{k=1}^N w_k = 1$.

The clusters represented by ABs have spherical shapes, they overlap and their sizes are limited by cross-reactivity thresholds. The number of clusters is here equal to the number of learning patterns, and the means of clusters in the pattern spaces X and Y (paratopes ABx and ABy) are fixed – they lie on the learning patterns.

The cross-reactivity thresholds, determining the cluster sizes, are tuned to the training data in the immune memory learning process. In results the clusters in the space X correspond to compact clusters in the space Y , and vice versa. It leads to more accurate mapping $X \rightarrow Y$. The model has four parameters – error thresholds (δ_y and ϵ_x) and parameters tuning the cross-reactivity thresholds (b and c). Increasing the values of these parameters imply an increase in size of clusters, an increase of the model bias and reduction of its variance.

The training routine is deterministic, which means fast learning process. The immune memory learning needs only one pass of the training data. The runtime complexity of the training routine is $O(N^2n)$. The most costly operation is the distance calculation between each ABs and AGs . The runtime complexity of the forecasting procedure is also $O(N^2n)$.

4 Application Example

The described above AIS was applied to the next day electrical load curve forecasting. Short-term load forecasting plays a key role in control and scheduling of power systems and is extremely important for energy suppliers, system operators, financial institutions, and other participants in electric energy generation, transmission, distribution, and markets.

The series studied in this paper represents the hourly electrical load of the Polish power system from the period 2002-2004. This series is shown in Fig. 1. The time series was divided into training and test parts. The test set contained 31 pairs of patterns from July 2004. The training set contained patterns from the period from 1 January 2002 to the day preceding the day of forecast.

For each day from the test part the separate immune memory was created using the training subset containing AGy representing days of the same type (Monday, ..., Sunday) as the day of forecast and paired with them AGx representing the preceding days (e.g. for forecasting the Sunday load curve, model learns from AGx representing the Saturday patterns and AGy representing the Sunday patterns). This adaptive routine of model learning provides fine-tuning its parameters to the changes observed in the current behavior of the time series.

The distance between ABs and AGs was calculated using Euclidean metric. The patterns were defined using (1).

The model parameters were determined using the grid search method on the training subsets in the local version of leave-one-out procedure. In this procedure not all patterns are successively removed from the training set but only the k -nearest neighbors of the test x -pattern (k was arbitrarily set to 5). As a result, the model is

optimized locally in the neighborhood of the test pattern. It leads to a reduction in learning time.

In the grid search procedure the parameters were changed as follows: (i) $\delta_y = 1.00, 1.25, \dots, 3.00, \epsilon_x = 1.00, 1.25, \dots, \delta_y$, at the constant values of $b = c = 1$, and (ii) $b = c = 0, 0.2, \dots, 1.0$, at the optimal values of δ_y and ϵ_x determined in point (i).

It was observed that at the lower values of δ_y and c many validation x-patterns remain unrecognized. If $\delta_y \geq 2.25$ and $c = 1$ approximately 99% of the validation x-patterns are detected by ABx. Increasing δ_y above 2.25 results in increasing the validation error. Minimum error was observed for $\delta_y = 2.25, \epsilon_x = 1.75$ and $b = c = 1$.

The forecast results are shown in Fig. 2 and 3. $MAPE$ for the test part of time series was 0.92 and its standard deviation – 0.72.

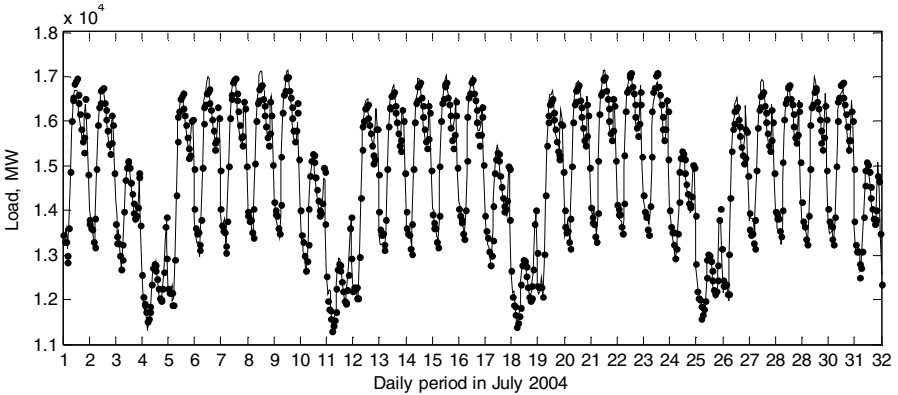


Fig. 2. Test part of the time series – July 2004 (solid lines) and its forecast (dots)

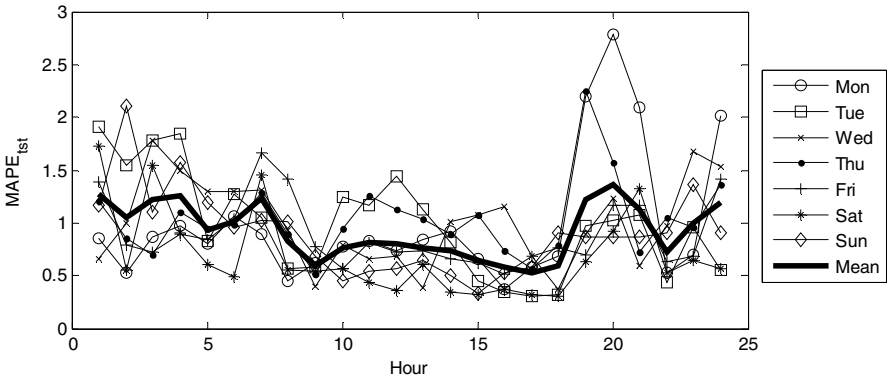


Fig. 3. $MAPE_{tst}$ for each day type and hour of the daily period

5 Conclusion

The proposed forecasting method belongs to the class of similarity-based models. These models are based on the assumption that, if patterns of the time series sequences are similar to each other, then the patterns of sequences following them are similar to each other as well. It means that patterns of neighboring sequences are staying in a certain relation, which does not change significantly in time. The more stable this relation is, the more accurate forecasts are. This relation can be shaped by proper pattern definitions and strengthened by elimination of outliers.

The idea of using AIS as a forecasting model is a very promising one. The immune system has some mechanisms useful in the forecasting tasks, such as an ability to recognize and to respond to different patterns, an ability to learn, memorize, encode and decode information.

Unlike other clustering methods used in forecasting models [1], [6], the proposed AIS forms clusters taking into account the forecast error. The cluster sizes are tuned to the data in such a way to minimize the forecast error. Due to the deterministic nature of the model results are stable and the learning process is rapid.

The disadvantage of the proposed immune system is limited ability to extrapolation. Regions without the antigens are not represented in the immune memory. However, a lot of models, e.g. neural networks, have problems with extrapolation.

Acknowledgments. The study was supported by the Research Project N N516 415338 financed by the Polish Ministry of Science and Higher Education.

References

1. Dudek, G.: Similarity-based Approaches to Short-Term Load Forecasting. In: Forecasting Models: Methods and Applications, pp. 161–178. iConcept Press (2010), http://www.iconceptpress.com/site/download_publishedPaper.php?paper_id=100917020141
2. Dudek, G.: Artificial Immune System for Short-term Electric Load Forecasting. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS(LNAI), vol. 5097, pp. 1007–1017. Springer, Heidelberg (2008)
3. Hart, E., Timmis, J.: Application Areas of AIS: The Past, the Present and the Future. *Applied Soft Computing* 8(1), 191–201 (2008)
4. Perelson, A.S., Weisbuch, G.: Immunology for Physicists. *Rev. Modern Phys.* 69, 1219–1267 (1997)
5. De Castro, L.N., Timmis, J.: Artificial Immune Systems as a Novel Soft Computing Paradigm. *Soft Computing* 7(8), 526–544 (2003)
6. Lendasse, A., Verleysen, M., de Bodt, E., Cottrell, M., Gregoire, P.: Forecasting Time-Series by Kohonen Classification. In: Proc. the European Symposium on Artificial Neural Networks, Bruges, Belgium, pp. 221–226 (1998)

Knowledge-Based Pattern Recognition Method and Tool to Support Mission Planning and Simulation

Ryszard Antkiewicz, Andrzej Najgebauer, Jarosław Rulka, Zbigniew Tarapata,
and Roman Wantoch-Rekowski

Military University of Technology, Cybernetics Faculty,
Gen. S. Kaliskiego Str. 2, 00-908 Warsaw, Poland
{rantkiewicz, anajgebauer, jrulka, ztarapata, rekowski}@wat.edu.pl

Abstract. This paper presents a model, method and computer tool for military mission planning. The actions on the battlefield should be planned on the basis of reconnaissance (decision situation recognition) and the possibility of own troops action and counteractions of the enemy. A course of action (CoA) should be verified and we propose a special tool for recommendation of CoA. Therefore, the pattern recognition method for identification of the decision situation is presented in the paper. The decision situation being identified is a basis for choosing CoA because with each decision situation a few typical CoA frames are connected. There is also a short description of the deterministic simulator, which is described on the basis of combat model, taking into account complimentary processes of firing and movement. The toolset presented with the CoA editor, military equipment, simulator manager and multicriteria estimation of CoA.

Keywords: Decision situation, pattern recognition, decision support system, military applications, mission planning and simulation.

1 Introduction

The typical military decision planning process contains the following steps:

- Estimation of power of own and opposite forces, terrain, and other factors, which may influence on a task realization,
- Identification of a decision situation,
- Determination of decision variants (Course of Actions, CoA),
- Variants (CoA) evaluation (verification),
- Recommendation of the best variant (CoA) of the above-stated points, which satisfy the proposed criteria.

Simulation and verification of course of actions (CoA) is considered in many systems and aspects [2], [4], [6], [7], [9], [14]. The most important step of decision planning process is an identification of a decision situation problem: this problem is that we must find the most similar battlefield situation (from earlier defined or ensuing situations, e.g. in a battlefield situation knowledge) to current one. Afterwards, the decision situation being identified is a basis for choosing CoA, because with each decision situation a few typical CoA frames are connected. In this paper we present the model of the decision situation and the method of finding the most similar decision situation to the current

one. Each decision situation pattern is described by eight parameters (section 2). The presented tool (called CAVaRS, described in details in [3]) allows to prepare knowledge base of decision situations patterns with values of characteristic parameters. It also allows to fix the best (nearest) decision situation located in the knowledge base, choose (or create) the best CoA and simulate selected CoAs.

The paper is organized as follows: in section 2 we have described the model and the pattern recognition method for the decision situation, section 3 contains the pattern recognition based tool for mission planning and simulation and in section 4 we have presented some examples of using this tool for military application.

2 Description of the Pattern Recognition Model and the Method for the Decision Situation

We define decision situations space as follows:

$$DSS = \{SD : SD = (SD_r)_{r=1,\dots,8}\} \quad (1)$$

Vector SD represents the decision situation, which is described by the following eight elements: SD_1 - command level of opposite forces, SD_2 - type of task of opposite forces (e.g. attack, defence), SD_3 - command level of own forces, SD_4 - type of task of own forces (e.g. attack, defence), SD_5 - net of squares as a model of activities (terrain) area $SD_5 = \left[SD_{ij}^5 \right]_{\substack{i=1,\dots,SD_7 \\ j=1,\dots,SD_8}}$, $SD_{ij}^5 = (SD_{ij}^{5,k})_{k=1,\dots,7}$. The terrain square with the indices (i,j) each of the elements denotes: $SD_{ij}^{5,1}$ - the degree of the terrain passability, $SD_{ij}^{5,2}$ - the degree of forest covering, $SD_{ij}^{5,3}$ - the degree of water covering, $SD_{ij}^{5,4}$ - the degree of terrain undulating, $SD_{ij}^{5,5}$ - armoured power (potential) of opposite units deployed in the square, $SD_{ij}^{5,6}$ - infantry power (potential) of opposite units deployed in the square, $SD_{ij}^{5,7}$ - artillery power (potential) of opposite units deployed in the square, $SD_{ij}^{5,7}$ - coordinates of the square (i,j) , SD_6 - the description of own forces: $SD_6 = (SD_i^6)_{i=1,\dots,4}$, SD_1^6 - total armoured power (potential) of own units, SD_2^6 - total infantry power (potential) of own units, SD_3^6 - total artillery power (potential) of own units, SD_4^6 - total air fire support power (potential); SD_7 - the width of activities (interest) in an area (number of squares), SD_8 - the depth of activities (interest) in an area (number of squares).

The set of decision situations patterns are: $PDSS = \{PS : PS \in DSS\}$. For the current decision situation CS , we have to find the most similar situation PS from the set of patterns. Using the similarity measure function (5) we can evaluate distances between two different decision situations, especially the current and the pattern. We have determined the subset of decision situation patterns $PDSS_{CS}$, which are generally similar to the current situation CS , considering such elements like: task type, command level of own and opposite units and own units potential:

$$PDSS_{CS} = \{PS = (PS_i)_{i=1,..,6} \in PDSS : PS_i = CS_i, i = 1,..,4, dist_{potwl}(CS, PS) \leq \Delta Pot\} \tag{2}$$

where:

$$dist_{potwl}(CS, PS) = \max\{|CS_k^6 - PS_k^6|, k = 1,..,4\} \tag{3}$$

ΔPot - the maximum difference of own forces potential.

Afterwards, we formulated and solved the multicriteria optimization problem (4), which allow us to determine the most matched pattern situation (PS) to the current one (CS) from the point of view of terrain and military power characteristics:

$$Z = (PDSS_{CS}, F_{CS}, R_D) \tag{4}$$

where:

$$F_{CS} : PDSS_{CS} \rightarrow R^2 \tag{5}$$

$$F_{CS}(PS) = (dist_{ter}(CS, PS), dist_{pot}(CS, PS)) \tag{6}$$

$$dist_{ter}(CS, PS) = \sum_{k=1}^4 \lambda_k \cdot \left(\sum_{i=1}^I \sum_{j=1}^J (CS_{ij}^{5,k} - PS_{ij}^{5,k})^p \right)^{\frac{1}{p}} \tag{7}$$

$$\sum_{k=1}^4 \lambda_k = 1, \lambda_k > 0, k = 1,..,4 \tag{9}$$

$$dist_{pot}(CS, PS) = \sum_{k=5}^7 \mu_k \cdot \left(\sum_{i=1}^I \sum_{j=1}^J (CS_{ij}^{5,k} - PS_{ij}^{5,k})^p \right)^{\frac{1}{p}} \tag{9}$$

$$\sum_{k=5}^7 \mu_k = 1, \mu_k > 0, k = 5,..,7 \tag{10}$$

$$I = \min\{CS_7, PS_7\}, J = \min\{CS_8, PS_8\} \tag{11}$$

$$R_D = \left\{ \begin{array}{l} (Y, Z) \in PDSS_{CS} \times PDSS_{CS} : \\ dist_{ter}(CS, Y) \leq dist_{ter}(CS, Z) \wedge \\ dist_{pot}(CS, Y) \leq dist_{pot}(CS, Z) \end{array} \right\} \tag{12}$$

Parameters μ_k and λ_k describes the weights for components calculating the value of functions $dist_{ter}$ and $dist_{pot}$. The domination relation defined in (12) allows us to choose such a PS from $PDSS_{CS}$, which has the best value of $dist_{ter}$ and $dist_{pot}$, that is the most similar to CS (non-dominated PS from the R_D point of view). The idea of the identification of decision situation and CoA selection is presented on Fig. 1.

There are several methods of finding the most matched pattern situation to the current, which can be used. For example, in paper [13] a concept of multicriteria

weighted graphs similarity and its application for pattern matching of decision situations is considered. The approach extends known pattern recognition approaches based on graph similarity with two features: (1) the similarity is calculated as structural and non-structural (quantitative) in weighted graph, (2) choice of the most similar graph to graph representing pattern is based on a multicriteria decision. Application of the presented approach to pattern recognition of decision situations has been described in [1], [15] as well.

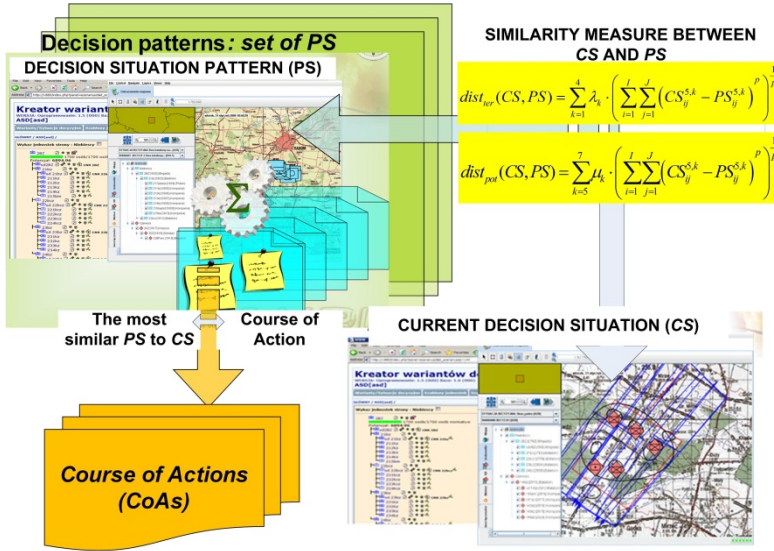


Fig. 1. The idea of identification of the decision situation and CoA selection

3 Description of the Pattern Recognition Based Tool for Mission Planning and Simulation

In this section we present tool, deterministic simulator called CAVaRS [3], which has been built at the Cybernetics Faculty of the Military University of Technology in Warsaw (Poland) and authors of this paper are members of the team, which has created them. CAVaRS may be used as a part of a Decision Support System (DSS), which supports Polish C4ISR systems or it may work as standalone.

The deterministic and discrete time-driven simulator CAVaRS models two-face land conflict of military units on the company/battalion level. The simulator is implemented in the JAVA language as an integrated part of some of the systems for CAXes. The model concerns a couple of processes of firing interaction and movement executed by a single military unit. These two complementary models use a terrain model described by a network of square areas, which aggregates movement characteristics with 200m×200m granularity (see Fig. 2). Examples of some square areas are showed below.

```

X: 15 to 15.25; Y: 52.167 to 52.333
dx = 0.003012; dy = 0.001792
Urban terrain, forest, swamp, lakes, rivers, trench, height, height diff.
X: 15, Y: 52.166666666667 00ff00000000 0.00 0.00
X: 15, Y: 52.168458781362 008300000000 0.00 0.00
X: 15, Y: 52.170250896073 000c00000000 0.00 0.00
...

```

The course of each process depends on many factors among them: terrain and weather conditions, states and parameters of weapons the units are equipped with, type of executed unit activities (attack, defence) and distance between opposing units.

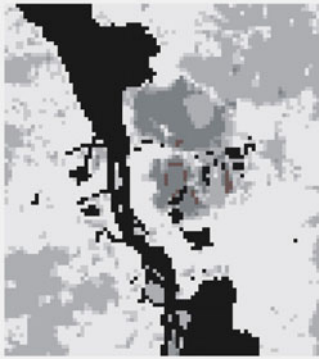


Fig. 2. Aggregate movement characteristics: the darker colour is the smallest terrain passability

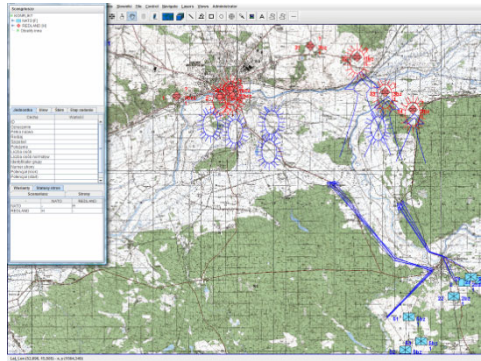


Fig. 3. Redeployment variant visualization in CAVaRS

Presented in Fig. 3 is the visualization of an exemplified redeployment variant in CAVaRS.

Scenarios of the variants of the military scenario in the Knowledge Base Editor of the simulator CAVaRS can be created in two ways: manually and half-automatic. In manual mode the variant can be built using military units templates stored in the CAVaRS database. In half-automatic mode the military scenario can be imported from others C3(4)ISR (e.g. C3ISR "Jaśmin") systems using MIP-DEM (Multilateral Interoperability Program - Data Exchange Mechanism) and MIP-JC3IEDM (Joint Consultation, Command and Control Information Exchange Data Model) integration database schema. The Knowledge Base Editor can import and transform data from MIP JC3IEDM standard data schema to CAVaRS data schema. This way is faster than manual mode, because all data of military units or at least most of them can be imported from other C3(4)ISR systems with detailed data such as unit position, equipment, weapons, etc.

4 A Practical Example of Using Tool

The example shows elements of knowledge base and the algorithm of nearest pattern situation searching.

The main element of the system is knowledge base, which consists of Decision Situations Patterns (DSP). Each DSP is connected to the set of Course of Actions (CoA). The example of two DSPs and their CoAs are presented below.

The first DSP (see Fig.4) is connected with two CoAs (see Fig.6 and Fig. 7). The second DSP is shown in the Fig. 8. Parameters have been fixed for each DSP. Fig. 5 shows the analyzed area of the opposite forces. Parameters of each DSP are kept in the knowledge base. Table 1 and Table 2 show values of DSP parameters.

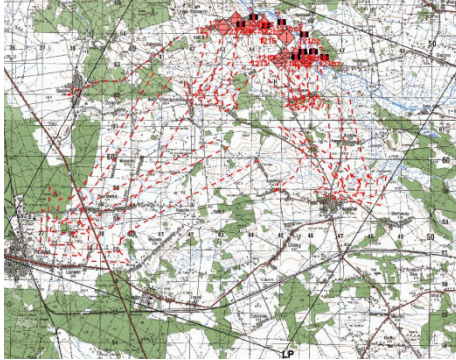


Fig. 4. Graphical representation of DSP 1

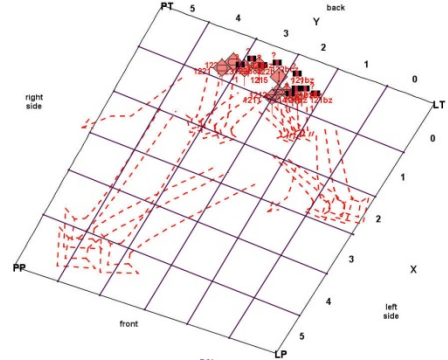


Fig. 5. DSP 1 – area of opposite forces

Table 1. Detailed values of DSP 1 parameters using notations from (1)

i	j	$SD_{ij}^{5,1}$	$SD_{ij}^{5,2}$	$SD_{ij}^{5,3}$	$SD_{ij}^{5,4}$	$SD_{ij}^{5,5}$	$SD_{ij}^{5,6}$	$SD_{ij}^{5,7}$
0	0	54%	1%	1%	0.069	0	0	0
0	1	44%	4%	1%	0.116	0	0	0
0	2	42%	15%	2%	0.186	0	17.46	94.13
0	3	45%	9%	4%	0.21	190	16.32	23.75
0	4	41%	8%	2%	0.252	80	5.2	0
0	5	42%	24%	1%	0.176	0	0	0
1	0	46%	23%	2%	0.12	0	0	0
1	1	54%	5%	1%	0.162	0	0	0
1	2	37%	15%	0%	0.231	0	26.98	140.8
1	3	47%	13%	0%	0.158	25	5.71	21.35
1	4	45%	10%	0%	0.177	25	1.62	0
1	5	35%	0%	34%	0.168	0	0	0
2	0	2%	0%	58%	0.096	0	0	0
2	1	7%	0%	54%	0.135	0	0	0
2	2	17%	0%	50%	0.183	0	0	0
2	3	11%	0%	38%	0.138	0	0	0
2	4	23%	0%	34%	0.162	0	0	0
2	5	51%	0%	29%	0.179	0	0	0
3	0	2%	0%	46%	0.168	0	0	0
...
5	5	25%	20%	0%	0.013	0	0	0

Coordinates of terrain area for DSP 1 (NW: north-west corner, NE: north-east corner, SW: south-west corner, SE: south-east corner):
NW (LP)=515556N 0213922E NE (PP)=515740N 0213053E
SW (LT)=520056N 0214431E SE (PT)=520254N 0213541E
Potential of own forces: mechanized 444; armoured 61.2; artillery 30; antiaircraft 0; other 0.

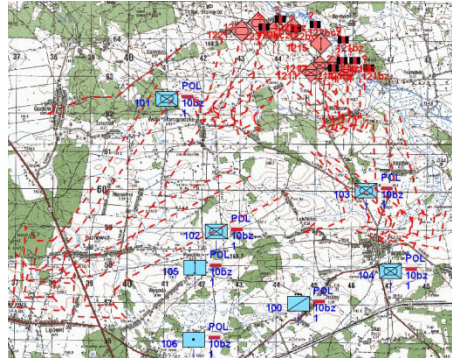
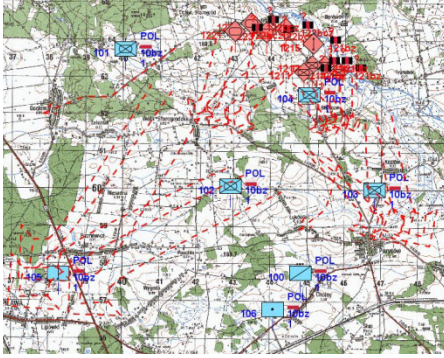


Fig. 6. Graphical representation of DSP 1, CoA 1 Fig. 7. Graphical representation of DSP 1, CoA 2

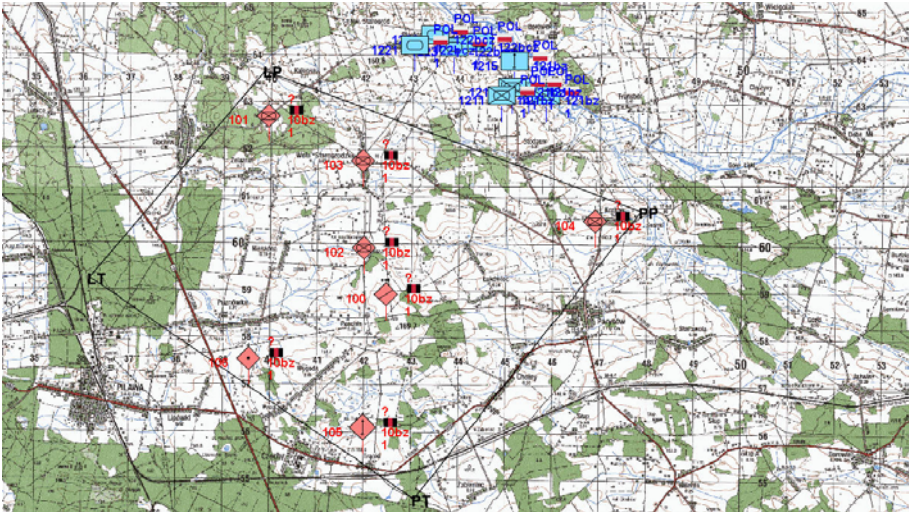


Fig. 8. Graphical representation of DSP 2

Coordinates of terrain area for DSP 2 (NW: north-west corner, NE: north-east corner, SW: south-west corner, SE: south-east corner):
NW (LP)=520120N 0213451E
NE (PP)=515943N 0214150E
SW (LT)=515858N 0213135E
SE (PT)=515625N 0213736E
Potential of own forces: mechanized 320; armoured 73.3; artillery 280; antiaircraft 0; other 0.

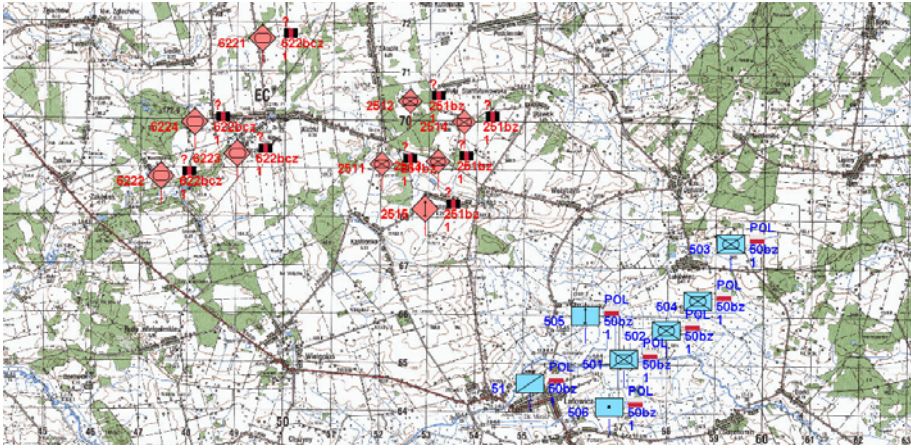


Fig. 9. Current situation (CS)

Table 2. Detailed values of DSP 2 parameters using notations from (1)

i	j	$SD_{ij}^{5,1}$	$SD_{ij}^{5,2}$	$SD_{ij}^{5,3}$	$SD_{ij}^{5,4}$	$SD_{ij}^{5,5}$	$SD_{ij}^{5,6}$	$SD_{ij}^{5,7}$
0	0	29%	93%	0%	0.01	0	0	0
0	1	55%	48%	0%	0.06	0	0	0
0	2	91%	1%	0%	0.04	8.62	4.49	0
0	3	84%	10%	0%	0.04	5.38	2.81	0
0	4	84%	11%	0%	0.03	0	5.85	27
0	5	76%	30%	0%	0.01	0	0.65	3
...
2	2	88%	0%	0%	0.03	13	1.44	0
2	3	84%	10%	0%	0.05	60	6.55	0
2	4	59%	44%	0%	0.07	6	0.6	0
2	5	77%	12%	0%	0.06	0	0	0
3	0	66%	33%	0%	0.09	0	0	0
3	1	83%	4%	0%	0.04	0	0	0
3	2	88%	3%	0%	0.02	6.5	0.72	0
3	3	80%	7%	0%	0.08	32.5	3.59	0
3	4	82%	1%	0%	0.1	0	0	0
3	5	81%	0%	0%	0.12	0	0	0
4	0	40%	74%	0%	0.08	66.9	7.39	0
4	1	62%	43%	0%	0.06	32.7	3.61	0
4	2	85%	1%	0%	0.05	93.6	10.4	0
4	3	70%	22%	0%	0.09	0	0	0
4	4	69%	9%	0%	0.15	0	0	0
4	5	87%	4%	0%	0.05	18.9	2.09	0
...
5	5	85%	6%	0%	0.05	85.1	9.41	0

Each DSP parameter of the current situation (see Fig.9) were calculated. The algorithm for finding the most similar pattern situation compares current situation parameters with each DSP from knowledge base using the method described in section 3. As a result the DSP1 has been fixed according to the $dist$ values (equation (7) and (9)) presented in Table 3 because: $dist_{pot}(CS, DSP1) < dist_{pot}(CS, DSP2)$ and $dist_{ter}(CS, DSP1) < dist_{ter}(CS, DSP2)$, hence DSP1 dominates DSP2 from the R_D (formula (12)) point of view.

Table 3. Detailed values of $dist$ parameters from (7) and (9)

DSP	$dist_{pot}(CS, DSP)$	$dist_{ter}(CS, DSP)$
DSP1	203.61	1.22
DSP2	222.32	1.47

5 Summary

Presented in this paper was the problem of using the knowledge base tool for mission planning. The presented tool (with knowledge base and pattern recognition method) was used during the practical experiment. The most important problem is to prepare the knowledge base with decision situation patterns and CoAs for each decision situation pattern. The specialized tool for knowledge base management was implemented and it allows to prepare decision situations patterns using GIS functions. The validation process of combat models is very difficult but it is possible to use such tools like simulation models calibrator and expert knowledge [5], [6], [10]. The construction of the simulation model enables the testing of different course of actions including ideas in the area of Network Enabled Capabilities [8].

Acknowledgments. This work was partially supported by the European Defence Agency project titled: “Asymmetric Threat Environment Analysis (through Asymmetric Engagement Modelling, Modelling of Impacts on Hearts & Minds, and Threat Scenario Generation from environmental factors) (ATHENA)” under the Call “Mission Planning/Training in an asymmetric environment” and “Secured Tactical Wireless Communications” (A-0676-RT-GC).

References

1. Antkiewicz, R., Najgebauer, A., Kulas, W., Pierzchała, D., Rulka, J., Tarapata, Z., Wantoch-Rekowski, R.: The Automation of Combat Decision Processes in the Simulation Based Operational Training Support System. In: Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA), Honolulu, Hawaii, USA, April 1-5 (2007) ISBN 1-4244-0698-6
2. Antkiewicz, R., Najgebauer, A., Kulas, W., Pierzchała, D., Rulka, J., Tarapata, Z., Wantoch-Rekowski, R.: Selected Problems of Designing and Using Deterministic and Stochastic Simulators for Military Trainings. In: 43rd Hawaii International Conference on System Sciences, pp. 5–8. IEEE Computer Society, USA (2010) ISBN 978-0-7695-3869-3

3. Antkiewicz, R., Koszela, J., Najgebauer, A., Pierzchała, D., Rulka, J., Tarapata, Z., Wantoch-Rekowski, R.: Fast CoA Verification and Recommendation using Tactical Level Land Battle Deterministic Simulator. In: UKSim 13th International Conference on Computer Modelling and Simulation, March 30 - April 1. IEEE Explore, Cambridge (2011) (in press) ISBN 978-0-7695-4376-5,
4. Barry, P.S., Koehler, M.T.K.: Leveraging agent based simulation for rapid course of action development. In: Proceedings of the 2005 Winter Simulation Conference, Orlando, FL, December 4 (2005)
5. Dockery, J., Woodcock, A.E.R.: The Military Landscape, Mathematical Models of Combat. Woodhead Publishing Ltd., Cambridge (1993)
6. Hofmann, H.W., Hofmann, M.: On the Development of Command & Control Modules for Combat Simulation Models on Battalion down to Single Item Level. In: Proceedings of the NATO/RTO Information Systems Technology Panel (IST) Symposium "New Information Processing Techniques for Military Systems", Istanbul, Turkey, October 9-11. NATO AC/329 (IST-017) TP/8, pp. 85-96 (2000)
7. Matthews, K.: Development of a Course of Action Simulation Capability, Command and Control Division Information Sciences Laboratory, DSTO-CR-0376, Edinburgh, Australia (2004)
8. Moffat, J.: Complexity Theory and Network Centric Warfare, CCRP Publication Series, Washington (2003) ISBN 1-893723-11-9
9. Najgebauer, A.: Polish Initiatives in M&S and Training. Simulation Based Operational Training Support System (SBOTSS) Zlocien. In: Proceedings of the ITEC 2004, London, UK (2004)
10. Przemieniecki, J.S.: Mathematical Methods in Defence Analysis. American Institute of Aeronautics and Astronautics, Inc., Washington, DC (1994)
11. Ross, K., Klein, G., Thunholm, P., Schmitt, J., Baxter, H.: The Recognition-Primed Decision Model. Military Review, 6-10 (July-August 2004)
12. Sokolowski, J.A.: Can a Composite Agent be Used to Implement a Recognition-Primed Decision Model? In: Proceedings of the Eleventh Conference on Computer Generated Forces and Behavioral Representation, Orlando, FL, May 7-9, pp. 473-478 (2002)
13. Tarapata, Z.: Multicriteria weighted graphs similarity and its application for decision situation pattern matching problem. In: Proceedings of the 13th IEEE/IFAC International Conference on Methods and Models in Automation and Robotics, Szczecin, Poland, August 27-30, pp. 1149-1155 (2007) ISBN 978-83-751803-3-6
14. Tarapata, Z.: Automatization of decision processes in conflict situations: modelling, simulation and optimization. In: Arreguin, J.M.R. (ed.) Automation and Robotics, pp. 297-328. I-Tech Education and Publishing, Vienna (2008) ISBN 978-3-902613-41-7
15. Tarapata, Z., Chmielewski, M., Kasprzyk, R.: An Algorithmic Approach to Social Knowledge Processing and Reasoning Based on Graph Representation – A Case Study. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) Intelligent Information and Database Systems. LNCS, vol. 5991, pp. 93-104. Springer, Heidelberg (2010)

Secure UHF/HF Dual-Band RFID : Strategic Framework Approaches and Application Solutions

Namje Park

Department of Computer Education, Teachers College, Jeju National University,
61 Iljudong-ro, Jeju-si, Jeju-do, 690-781, Korea
namjepark@jejunu.ac.kr

Abstract. In the mobile RFID (Radio-Frequency Identification) environment, scanning RFID tags which are personalized can bring some privacy infringement issues. In spite of the case that private information is not stored in those tags, one can identify entities, analyze their preferences, and track them by collecting data related with the tags and aggregating it. Especially, it might be more serious at the point of that data collection may be done not only by enterprises and government, but also by individuals. In this paper, we describe privacy infringements for the mobile RFID service environment. The proposed framework provides a means for securing the stability of mobile RFID services by suggesting personal policy based access control for personalized tags.

Keywords: RFID, Security, Mobile RFID, Privacy, hospital, Healthcare.

1 Introduction

RFID (Radio Frequency Identification) technology is currently widely used for supply chain management and inventory control. Furthermore, RFID is recognized as the vehicle to realize the ubiquitous environment. Though the Radio Frequency Identification (RFID) technology is being actively developed with a great deal of effort to generate a global market, it also is raising fears of its role as a 'Big Brother'. So, it is necessary to develop technologies for information and privacy protection as well as promotion of markets (e.g., technologies of tag, reader, middleware, etc.) The current excessive limitations to RFID tags and readers make it impossible to apply present codes and protocols. The technology for information and privacy protection should be developed in terms of general interconnection among elements and their characteristics of RFID to such technology that meets the RFID circumstances.

The typical architecture for RFID is composed of RFID tag, which is embedded or attached to an object, and the RFID reader and IS (Information Services) server. The RFID reader reads the code in the RFID tag and recognizes the meaning of the code via communicating with the IS server via proper communication network. This is the typical architecture defined by EPCglobal [1,2,3]. The RFID reader can be a type of stationary or mobile. If the RFID reader is mobile, then we can have more applications than the stationary RFID reader.

While common RFID technologies are used in B2B (Business to Business) models like supply channels, distribution, logistics management, mobile RFID technologies are used in the RFID reader attached to an individual owner's cellular phone through which the owner can collect and use information of objects by reading their RFID tags; in case of corporations, it has been applied mainly for B2C (Business to Customer) models for marketing. Though most current mobile RFID application services are used in fields like the search of movie posters and provision of information in galleries where less security is required, they will be expanded to and used more frequently in such fields as purchase, medical care, electrical drafts, and so on where security and privacy protection are indispensable. A method to solve the problem of the mobile RFID service has been studied by researchers [6,8,9,12].

In this paper, we explain UHF/HF RFID technology based on EPC and analyze threats of the mobile RFID service. We propose privacy protection service framework based on a user privacy policy. The proposed framework provides a means for securing the stability of mobile RFID services by suggesting personal privacy-policy based access control for personalized tags. This is new technology to mobile RFID and will provide a solution for protecting absolute confidentiality from basic tags to user's privacy.

2 Strategic Security Framework Architecture

This technology is aimed at RFID application services like authentication of tag, reader, and owner, privacy protection, and non-traceable payment system where stricter security is needed [6,11,12,14,15].

- Approach of Platform Level

This technology for information portal service security in offering various mobile RFID applications consists of application portal gateway, information service server, terminal security application, payment server, and privacy protection server and provides a combined environment to build a mobile RFID security application service easily.

- Approach of Protocol Level

It assists write and kill passwords provided by EPC (Electronic Product Code) Class1 Gen2 for mobile RFID tag/reader and uses a recording technology preventing tag tracking. Information technology solves security vulnerability in mobile RFID terminals that accept WIPI as middleware in the mobile RFID reader / application part and provides E2E (End-to-End) security solutions from the RFID reader to its applications through WIPI based mobile RFID terminal security / code treatment modules.

- Approach of Privacy Level

This technology is intended to solve the infringement of privacy, or random acquisition of personal information by those with RFID readers from those with RFID attached objects in the mobile RFID circumstance except when taking place in companies or retail shops that try to collect personal information. The main assumptions are privacy in the mobile RFID circumstance when a person holds a tag attached object and both information on his/her personal identity (reference number, name, etc.) and the tag's information of the commodity are connected. Owners

have the option to allow access to any personal information on the object's tag by authorized persons like a pharmacist or doctor but limit or completely restrict access to unauthorized persons.

3 Implementation of Application Solutions

3.1 Implementation of Customized Healthcare Service

We implemented the proposed system for tracking patient care at a hospital. Context-relevant information is important in a ubiquitous computing environment for providing medical care. Different user policies are necessary for patient tags and product tags in EPCglobal's enterprise application. This ubiquitous sharing system for medical information poses a serious threat to the privacy of personal medical information such location, health, and clinical history. Standards such as Health Level Seven (HL7) do not allow customization and do not include rigorous privacy mechanisms. Therefore, we propose a mechanism that manages privacy policy in a user-centric manner for ubiquitous medical care. It is flexible, secure, and can be integrated with a cryptographic algorithm for mitigating the aforementioned problems.

3.2 Design M-RPS Based Customized Service

In a hospital, tags can be used for asset management for location finding. Patient tags are effective in preventing medical accidents, but must be properly designed and constructed to avoid massive collateral damage to user privacy. Hence, we define three-step privacy-aware service architecture for our mobile RFID-based medical application service [14,15,16]. The first step is setting the default level of access control over patient information in the default policy. The second step is user-controllable profile-based privacy protection, and the third step is auditable privacy management. Furthermore, we introduce a new RFID-based service model and mobile phone application.

The mobile RFID reader requests for information related to a tag attached to a patient from the backend IS via the middleware system. The mechanism allows individuals to control who can access their personal information. For privacy management, we apply the proposed profile-based privacy management architecture by the addition of a privacy bit to the tag, which is a simple and cost effective mechanism. The privacy bit is the only reference for the privacy service. The medical RFID information server check the privacy guaranteed service or not from the privacy policy. To illustrate how the privacy policy works on the IS, let us consider its use in the application and content information system of the service provider. The privacy level is stored in M-RPS. The RFID code format for the application is defined in the mobile RFID application data format as standard. The default privacy level follows the privacy applied standard of each application service; and if there is no standard, the privacy level is determined based on the results of a privacy impact assessment. The privacy level consists of a 10-tuple of information, where 'L = L1, L2,...L10' as the default privacy policy. It also protected by a secure tag area and privacy server. We also define privacy weights for medical information, as shown in Table 1.

Table 1. Examples of a Default Privacy Weight Level

Privacy related People	Privacy Weight	Privacy Information
Doctor	L4 ~ L9	Medical History Treatment Information
Nurse	L4 ~ L9	Medical History Treatment Information
General Doctor	L3 ~ L7	Medical Treatment
General Nurse	L3 ~ L7	Medical Treatment
Family	L2 ~ L6	Medical Tracking Information
Emergency Agency	L2 ~ L6	Medical Tracking Information
Others	L1	All cut off

Classify the personal medical information by patient’s policy and make personal’s profile. The patient can control his privacy level. Encrypted information can be transferred between the hospital and an emergency transportation service in XML format with security (WS Security) and also can be subject to the standard access control technology for Web services (XACML).

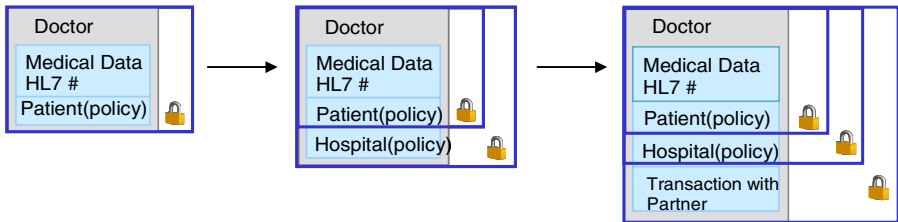


Fig. 1. Electronic signature and authentication

In the proposed hospital data management system, RFID-tagged medical card are given to patients on registration. Patients with sensitive conditions, for example, heart disease or cerebral hemorrhage, can use the medical card to rapidly provide medical history that can be used for fast application of first aid. Further, biosensors can be incorporated to provide real-time data to the doctor for each specific patient. The RFID patient tags also can be used to verify patient identity to ensure the correct treatment is administered. Thus, the system allows chartless service.

3.3 Implementation

The hospital generated an initial set of control data, which included the patient code, medical ID, and related information. The default privacy level was used and the patient was not allowed to control security policy. In order to provide authentication and

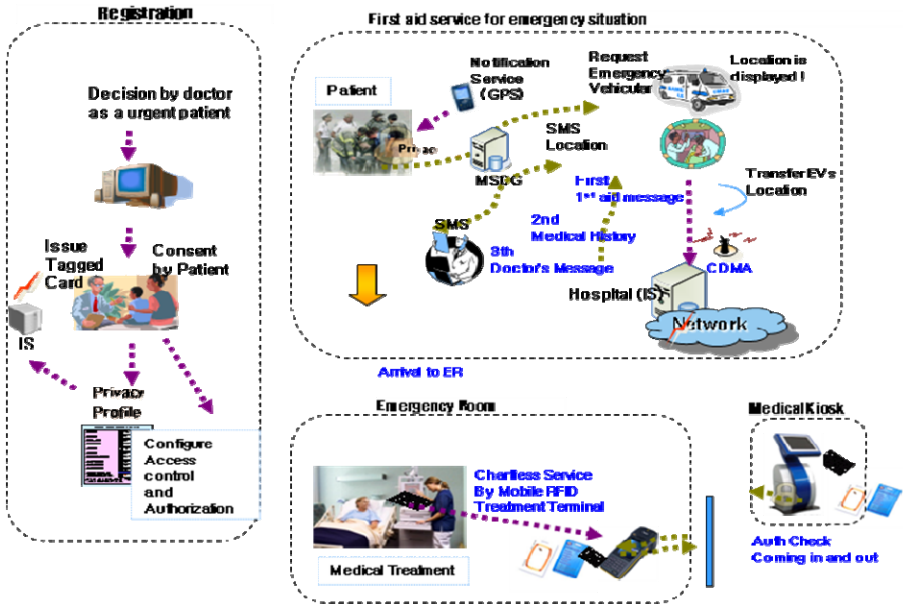


Fig. 2. Proposed customized ubiquitous hospital model

privacy interface to patient as an agent in medical discovery gateway and hospital's information server system. Essentially, each bit of sensitive data was initially classified by the default privacy weight, which was then modified by the end user's detailed policy. The user-controllable privacy policy in this system evaluation is considered a basic part of RFID privacy management. The compatibility and scalability may be limited, which will hamper system migration, but the mechanism is suitable for policy based privacy control. The proposed privacy management mechanism was implemented in an actual medical emergency room, including a networked medical information RFID kiosk, RFID networked emergency rescue system, and medical examination service, as shown in Fig. 3. There is some approach applying the RFID to medicine and hospital. From above, proposed privacy scheme has advantages in custom centric approach aspect for constructing a privacy aware ubiquitous medical system.

3.4 Performance Evaluation

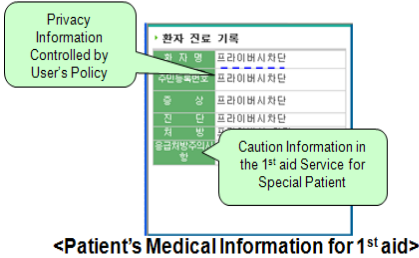
We tested the tag and middleware platform to verify the performance of the proposed framework based on the implemented UHF/HF mobile RFID reader. The mobile RFID performance test was conducted as follows:

- The read range was measured at that the power corresponding to the maximum read range (within 20–30 dBm of the maximum antenna power, less than 6 dBi antenna gain).

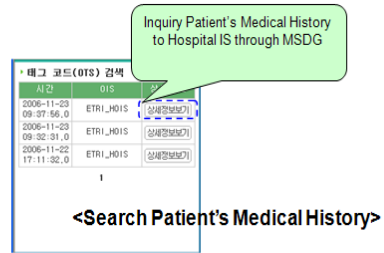
A frequency-hopping spread spectrum (FHSS)-type reader was tested with more than 15 hopping channels, less than 0.4 s of channel possession, and anti-collision.



<Emergency Transportation IS>



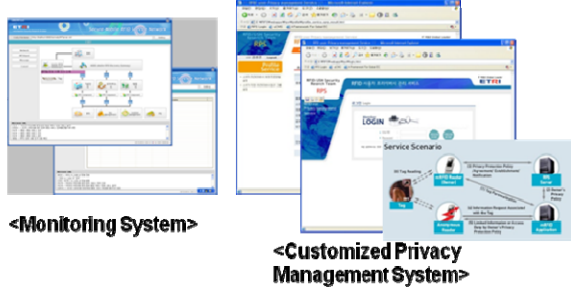
<Patient's Medical Information for 1st aid>



<Search Patient's Medical History>



<Kiosk>



<Monitoring System>

<Customized Privacy Management System>

<Mobile RFID Reader>

Fig. 3. Medical examination with proposed system

- The reader sent modulated signals while complying with a multiple-reader spectrum mask.
- After the maximum UII (mCode, micro-mCode or mini-mCode) and application data (only if the tags supporting application data are used: ISO 18000-6C tag: AD set of more than 26 bytes, ISO 18000-6B tag: AD set of more than 50 bytes) that the tag c

an support are recorded in the {P+OID+O} format in the tag memory in accordance with compression regulations (UII: Application defined (000), application data: Application defined (000), [UTF-8 String type Value-UTF-8, Decimal Numeric Character String type Value-Numeric string compaction]), the read range is measured under the condition that the UII and application data are locked.

- The UII recognition (or UII recognition and application data reading) method applied to the reader's code parsing in a terminal application should be different from that in the network.
 - When code parsing is executed in the terminal application, the parsing processing function verifies the UII recognition with URN or FQDN in the reader application on display, maps the pair of code & application data for each tag (single tag, multiple tags) and displays the application data under the equivalent UII.
 - When code parsing is conducted in a network, the UII set (ISO 18000-6C: PC field+DSFID, ISO 18000-6B: including 12~15 byte + DSFID) is confirmed in the reader application display, the code and application are mapped as a pair for each tag (single tag, multiple tags) and the application data is displayed under the equivalent UII set [ISO 18000-6C: PC field + DSFID, ISO 18000-6B: including 12~15 byte + DSFID].
- Terminals that have passed the mobile RFID standard conformance test and interoperability test and satisfy the standard performance requirements (a maximum read range of more than 40 cm) should be selected as standard terminals (reader or tag).
- The cross-sectional area of the antenna pattern for a label tag and metal tag in the test should be less than 7,500 mm² or a standard tag should be used for the test.
- The tag manufacturer should provide a test authority with at least 200 tags. The tags to be used for the tests are selected by random sampling. Then, the read range is measured after aging conducted twice under temperature/humidity conditions as shown in the following figure.
 - Tags with a proven read range of 30 cm should be used as standard tags for measuring the read range of the readers.

4 Conclusion

RFID technology will evolve into an intellectual ubiquitous environment by mounting an RFID tag to every object, automatically detecting the information of the surrounding environment, and interconnecting them through the network. Especially, RFID technology requires technological security measures as it is vulnerable to privacy infringement like counterfeiting, falsification, camouflage, tapping, and global positioning through illintentioned attack. Therefore, it is necessary to enact laws and regulations that can satisfy all consumer protection organizations that are sensitive to individual privacy infringement, and develop and apply the security technology that can apply such laws and regulations to all products.

The mobile RFID technology is being actively researched and developed throughout the world and more efforts are underway for the development of related service technologies. Though legal and institutional systems endeavor to protect privacy and

encourage protection technologies for the facilitation of services, the science and engineering world also has to develop proper technologies. Seemingly, there are and will be no perfect security / privacy protection technology. Technologies proposed in this paper, however, would contribute to the development of secure and reliable network RFID circumstances and the promotion of the mobile RFID market.

Acknowledgments. This paper is extended from a conference paper presented at IFIP TC6 11th International Conference on Personal Wireless Communications. The author is deeply grateful to the anonymous reviewers for their valuable suggestions and comments on the first version of this paper.

References

1. Park, N.: Implementation of Terminal Middleware Platform for Mobile RFID computing. *International Journal of Ad Hoc and Ubiquitous Computing* (2011)
2. Kim, Y., Lee, J., Yoo, S., Kim, H.: A Network Reference Model for B2C RFID Applications. In: *Proceedings of ICACT* (2006)
3. Chae, J., Oh, S.: Information Report on Mobile RFID in Korea. ISO/IEC JTC 1/SC 31/WG 4 N 0922, Information paper, ISO/IEC JTC 1 SC 31 WG4 SG 5 (2005)
4. Park, N., Song, Y., Won, D., Kim, H.: Multilateral Approaches to the Mobile RFID Security Problem Using Web Service. In: Zhang, Y., Yu, G., Hwang, J., Xu, G. (eds.) *APWeb 2008*. LNCS, vol. 4976, pp. 331–341. Springer, Heidelberg (2008)
5. Park, W., Lee, B.: Proposal for participating in the Correspondence Group on RFID in ITU-T. Information Paper. *ASTAP Forum* (2004)
6. Park, N., Kwak, J., Kim, S., Won, D., Kim, H.: WIPI Mobile Platform with Secure Service for Mobile RFID Network Environment. In: Shen, H.T., Li, J., Li, M., Ni, J., Wang, W. (eds.) *APWeb Workshops 2006*. LNCS, vol. 3842, pp. 741–748. Springer, Heidelberg (2006)
7. Park, N., Kim, H., Kim, S., Won, D.: Open Location-Based Service Using Secure Middleware Infrastructure in Web Services. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) *ICCSA 2005*. LNCS, vol. 3481, pp. 1146–1155. Springer, Heidelberg (2005)
8. Chug, B., et al.: Proposal for the study on a security framework for mobile RFID applications as a new work item on mobile security. ITU-T, COM17D116E, Q9/17, Contribution 116, Geneva (2005)
9. Park, N., Kim, H.W., Kim, S., Won, D.H.: Open Location-Based Service Using Secure Middleware Infrastructure in Web Services. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) *ICCSA 2005*. LNCS, vol. 3481, pp. 1146–1155. Springer, Heidelberg (2005)
10. Lee, J., Kim, H.: RFID Code Structure and Tag Data Structure for Mobile RFID Services in Korea. In: *Proceedings of ICACT* (2006)
11. Park, N., Kim, S., Won, D.: Privacy Preserving Enhanced Service Mechanism in Mobile RFID Network. In: *ASC. Advances in Soft Computing*, vol. 43, pp. 151–156. Springer, Heidelberg (2007)
12. Park, N.: Security scheme for managing a large quantity of individual information in RFID environment. In: Zhu, R., Zhang, Y., Liu, B., Liu, C. (eds.) *ICICA 2010*. CCIS, vol. 106, pp. 72–79. Springer, Heidelberg (2010)

13. Park, N., Kim, S., Won, D., Kim, H.: Security Analysis and Implementation Leveraging Globally Networked RFIDs. In: Cuenca, P., Orozco-Barbosa, L. (eds.) PWC 2006. LNCS, vol. 4217, pp. 494–505. Springer, Heidelberg (2006)
14. Park, N., Lee, H., Kim, H., Won, D.: A Security and Privacy Enhanced Protection Scheme for Secure 900MHz UHF RFID Reader on Mobile Phone. In: IEEE Tenth International Symposium on Consumer Electronics, ISCE 2006, pp. 1–5. IEEE, Los Alamitos (2006)
15. Park, N., Kim, H., Chung, K., Sohn, S.: Design of an Extended Architecture for Secure Low-Cost 900MHz UHF Mobile RFID Systems. In: IEEE Tenth International Symposium on Consumer Electronics, ISCE 2006, pp. 1–6. IEEE, Los Alamitos (2006)
16. Park, N., Gadh, R.: Implementation of Cellular Phone-based Secure Light-Weight Middleware Platform for Networked RFID. In: 28th International Conference on Consumer Electronics, ICCE 2010, pp. 495–496. IEEE, Los Alamitos (2010)
17. Park, N.: Reliable System Framework leveraging Globally Mobile RFID in Ubiquitous Era. In: Ph.D. Thesis. Sungkyunkwan University, South Korea (2008)
18. Lee, H., Kim, J.: Privacy Threats and Issues in Mobile RFID. In: Proceedings of the First International Conference on Availability, Reliability and Security, vol. 1 (2006)
19. MIC (Ministry of Information and Communication) of Korea (2005) RFID Privacy Protection Guideline. MIC Report Paper (2005)
20. Mobile RFID Forum of Korea: WIPI C API Standard for Mobile RFID Reader. Standard Paper (2005)
21. Mobile RFID Forum of Korea: WIPI Network APIs for Mobile RFID Services. Standard Paper (2005)
22. Mobile RFID Forum of Korea: Mobile RFID Code Structure and Tag Data Structure for Mobile RFID Services. Standard Paper (2005)
23. Mobile RFID Forum of Korea: Access Right Management API Standard for Secure Mobile RFID Reader, MRFS-4-03. Standard Paper (2005)
24. Mobile RFID Forum of Korea: HAL API Standard for RFID Reader of Mobile Phone, Standard Paper (2005)
25. Park, N., Kim, Y.: Harmful Adult Multimedia Contents Filtering Method in Mobile RFID Service Environment. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010. LNCS(LNAI), vol. 6422, pp. 193–202. Springer, Heidelberg (2010)
26. Park, N., Song, Y.: AONT Encryption Based Application Data Management in Mobile RFID Environment. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010. LNCS(LNAI), vol. 6422, pp. 142–152. Springer, Heidelberg (2010)
27. Park, N., Song, Y.: Secure RFID Application Data Management Using All-Or-Nothing Transform Encryption. In: Pandurangan, G., Anil Kumar, V.S., Ming, G., Liu, Y., Li, Y. (eds.) WASA 2010. LNCS, vol. 6221, pp. 245–252. Springer, Heidelberg (2010)

Kernel PCA in Application to Leakage Detection in Drinking Water Distribution System

Adam Nowicki and Michał Grochowski

Faculty of Electrical and Control Engineering, Gdansk University of Technology,
Narutowicza str. 11/12, 80-233 Gdansk, Poland
adam.j.nowicki@gmail.com, m.grochowski@ely.pg.gda.pl

Abstract. Monitoring plays an important role in advanced control of complex dynamic systems. Precise information about system's behaviour, including faults detection, enables efficient control. Proposed method- Kernel Principal Component Analysis (KPCA), a representative of machine learning, skilfully takes full advantage of the well known PCA method and extends its application to nonlinear case. The paper explains the general idea of KPCA and provides an example of how to utilize it for fault detection problem. The efficiency of described method is presented for application of leakage detection in drinking water systems, representing a complex and distributed dynamic system of a large scale. Simulations for Chojnice town show promising results of detecting and even localising the leakages, using limited number of measuring points.

Keywords: machine learning, kernel PCA, fault detection, monitoring, water leakage detection.

1 Introduction

Several studies aims at estimating the losses in drinking water distribution systems (DWDS). Even though they differ with respect to the measurement methods and hence, are difficult to compare, the results are always alarming; European Commission studies show that they can be as high as 50% in certain areas of Europe. The losses can be considered as a difference between the volume of the water delivered to the system and the volume of authorized consumption (nonrevenue water-NRW). The World Bank estimates the worldwide NRW volume to be 48.6 billion m³/year [1] - most of it is caused by leakages. The most popular approach for detecting the leakages is acoustic based method [2-3]; overview of other methods can be found in [4-5]. The approach presented in this paper makes use of the fact that the appearance of the leakage can be discovered through analysis of flow and pressure measurements.

Water supply networks usually occupy large territories and are often subject to local disturbances which have limited effect on the remaining part of the network. This motivates building a number of local models rather than a single model of the entire network. The typical quantities measured are flows and pressures. The place the measurements are taken has an important impact on the efficiency

of the monitoring system. In order to reduce the cost of the whole system it is desirable to deploy the instruments in a concentrated manner – around pipe junctions (called nodes), preferably with as many pipes crossing as possible. Then, for a node with n pipes, $n+1$ measurements are available: n flows and a pressure. A model of such a node serves as a local model.

A variety of methods for fault detection can be applied for leakage detection problem. An extensive review of most common approaches can be found in [6]. Water distribution system is dynamic, complex, nonlinear system with varying parameters. Clearly, in this case the quantitative modelling is a very demanding task, while there is no straightforward solution for qualitative approach. Moreover, the values of flows and pressure measured in a real-life networks at a given node are proven to be highly repeatable on the daily basis with a predictable variations depending on the season. During the leakage the relationship between measurements is disturbed thus providing a fault symptom. These aspects motivates the choice of data-driven approach for a problem of leakage detection. This paper presents the results of employing the Kernel Principal Component Analysis to this problem. Results of application of other data-driven approaches for the leakage detection can be found in [5],[7].

2 Data-Driven Approach for Novelty Detection Using KPCA

Consider a single row vector of a size $[1 \times N]$ containing a set of N measurements taken at the same time, denoted as x_i :

$$x_i = [x_{i1} \quad x_{i2} \quad \cdots \quad x_{iN}] . \quad (1)$$

This vector belongs to N -dimensional space called input space R^N . The measured signals are assumed to be chosen so that the vector x_i determines operational state of the process, but not necessarily uniquely. It is sufficient that for any two measurements:

$$x_i \in NS, x_j \in FS: x_i \neq x_j , \quad (2)$$

where NS and FS correspond to data collected during normal and faulty state of the process, respectively. This means that a single vector x_i is capable of carrying a symptom of the fault.

When dealing with data-driven methods for a fault detection, there are two approaches to determine whether the fault has occurred: novelty detection and classification. This paper presents solution based on novelty detection, where the model is built using the data from NS set only – a training set with a number of data points significantly larger than the dimension of an input space and covering all operational states of the process. Then, the task of fault detection can be reduced to the problem of finding a decision boundary in N -dimensional input space that tightly encloses the training set. Hence, when previously unknown data is presented, the fault detection system is able to separate ordinary from novel patterns. If the NS data follows a continues linear pattern, PCA is a method of choice. This purely

statistical method uses hypersphere as a decision boundary [8]. Unfortunately, most of real world applications involves dealing with non-linear patterns. A remedy to this might be VQPCA: it uses a number of local PCA models which are built using Voronoi scheme. However, its application is restricted to the cases where pattern can be approximated with piecewise linear patterns and no determinism is required. A relatively new method that does not suffer from these drawbacks is the Kernel PCA, introduced in [9]. It can be considered as a non-linear extension of PCA that combines multivariate analysis and machine learning. Instead of looking for a linear relationship between the variables in the input space, all measurements are mapped into a higher dimensional feature space F through a non-linear mapping $\phi(\bullet)$:

$$x_i \rightarrow \phi(x_i) \quad i = 1, 2, \dots, N. \quad (3)$$

A subspace F_R is identified within the feature space where PCA is used. Linear patterns detected in this space corresponds to non-linear patterns in the input space R^N . The desirable size of F_R is such that it allows to capture only the general pattern of the data; normally F_R is of a higher dimension than R^N .

In a classical approach operations in large-dimensional spaces yields considerable workload since each vector is represented by a number of coordinates. Kernel PCA, which represents a group of so-called 'kernel methods', solves this problem using 'the kernel trick' described in [10]. For any algorithm which operates exclusively on inner product between data vectors, it is possible to substitute each occurrence of inner product with its kernel representation:

$$\langle \phi(x), \phi(y) \rangle = k(x, y). \quad (4)$$

Inner product can be interpreted as a similarity measure between data points: if the angle between two different vectors is small it means that both data points follows the same linear pattern. A value of $k(x,y)$ is calculated using chosen kernel function, which operates on data in input space R^N , but corresponds to the inner product between data in F , thus allowing to detect linear patterns there. This means that there is no need to carry out the mapping from the input space into the feature space, explicitly. Moreover, neither coordinates of the $\phi(x)$, nor even mapping $\phi(\bullet)$ is needed to be known - from this point of view it is the kernel function that defines this mapping $\phi(\bullet)$. The choice of a proper function can be motivated by specific domain knowledge – this enables to incorporate heuristics into the method. In order to check if new data follows the pattern discovered in a training set, a mechanism based on the reconstruction error may be used [8],[11]. This solution assumes that a subspace F_R of feature space F found during the training is suitable only for the data similar to the training set X . This means that for such a data, during the mapping $\phi: R^N \rightarrow F_R$ minimum information is lost and gives almost the same result as mapping $\phi: R^N \rightarrow F$, thus not producing large reconstruction error.

3 KPCA Model

Application of the Kernel PCA method for monitoring purposes is a two-step process: in the first step the model is built and then, in the second step this model is used to

determine the status of the system based on the latest measurements. In order to simplify the problem it is assumed that the model is non-adaptive which means that training set remains constant and therefore model is built only once.

Let X be the matrix containing normalized training set with data points $x_i, i=1,2,\dots,m$ given as row vectors in the input space $x_i \in \mathbb{R}^N$:

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{m,n} \end{bmatrix}. \tag{5}$$

Since this is novelty detection it is assumed that $x_i \in NS$ for $i=1,2\dots m$ (data set represents only normal operating states). In kernel methods all the information about the data is stored in Kernel matrix K which contains the value of the kernel function $k(x_i, x_j)$ calculated for each pair of vectors x_i from X . For gauss function:

$$K_{ij} = k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right), \quad \sigma > 0. \tag{6}$$

The value of the free parameter σ is chosen empirically. Since $K_{ij}=1$ and $K_{ji}=K_{ij}$, only elements above the diagonal need to be computed, which means that computation of Kernel matrix K of $[m \times m]$ size requires $\sum_{i=1}^{m-1} i$ evaluations of the kernel function.

Classic PCA requires that data is normalized, i.e. $\sum_{i=1}^m x_{ij} = 0$ for $j=1,2,\dots,n$. This is also the case when using Kernel PCA, but since data is expressed in terms of inner product, the normalization is applied indirectly, through Kernel matrix K . Each element of normalized Kernel matrix \tilde{K} can be expressed in terms of K :

$$\tilde{K}_{ij} = K_{ij} - \frac{1}{m} \sum_{r=1}^m K_{ir} - \frac{1}{m} \sum_{r=1}^m K_{rj} + \frac{1}{m^2} \sum_{r,s=1}^m K_{rs}. \tag{7}$$

A mapping $\tilde{\phi}(x) = \phi(x) - \phi_0$ which takes into account that the centre of the mass ϕ_0 of the training set X is moved to the origin of the coordinate system in the feature space F , is associated with centring procedure given in (7).

Normally, when applying classical PCA to the linear problems, eigenvectors v_i of the covariance matrix $C = \frac{1}{m-1} X^T X$ are searched for, since they define principal components. In Kernel PCA the algorithm is applied to data in the feature space F , so the primal PCA problem could be solved by finding eigenvectors of $C = \frac{1}{m-1} \tilde{\phi}^T(X) \tilde{\phi}(X)$, only the mapped data points $\tilde{\phi}(X)$ are not available explicitly. This problem is solved with different version of PCA, called dual PCA, which allows to compute eigenvectors v_i of $\tilde{\phi}^T(X) \tilde{\phi}(X)$ using $\tilde{\phi}(X) \tilde{\phi}^T(X) = \tilde{K}$ by:

$$v_i = \tilde{\phi}^T(X) \frac{u_i}{\sqrt{\lambda_i}}, \tag{8}$$

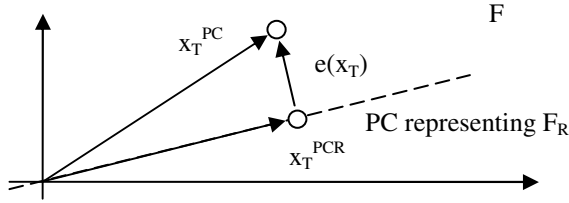


Fig. 1. Geometrical interpretation of the reconstruction error

where λ_i is i -th eigenvalue associated with i -th eigenvector u_i of \tilde{K} given as a column vector. Although $\tilde{\phi}(X)$ in (8) is not available explicitly, it will be possible to use v_i in this form later on.

It is worth noting that with primal PCA at most n eigenvectors for the covariance matrix C can be evaluated, while in Kernel PCA there can be evaluated as many as m eigenvectors that spans m -dimensional feature space F . Since it is always possible to construct a single hyperplane consisting of any m points in m -dimensional space, thus it is possible to find a linear relationship between all mapped points; however, this might lead to overfitting. Therefore only s eigenvectors corresponding to s largest eigenvalues are used, resulting in the subspace F_R that captures the general pattern in the data. The value of s is chosen empirically. These s eigenvalues are stored as column vectors in matrix V_R .

Having a model defined, it is possible to evaluate a reconstruction error $E(\bullet)$. For a new data point x_T , this error, denoted as $E(x_T)$, can be regarded as the squared distance $e(x_T)$ between the exact mapping of x_T into the feature space F and its projection onto the chosen subspace F_R . Let x_T^{PC} and x_T^{PCR} denote PC scores of x_T associated with the feature space F and reduced feature space F_R , respectively. Since principal components originates from the origin of coordinate system, using Pythagoras theorem (Fig. 1):

$$E(x_T) = \|e(x_T)\|^2 = \|x_T^{PC}\|^2 - \|x_T^{PCR}\|^2 . \tag{9}$$

The term $\|x_T^{PC}\|^2$ is a distance of the x_T from the origin of the coordinates in the feature space F and can be calculated using inner product calculation:

$$\begin{aligned} \|x_T^{PC}\|^2 &= \|\tilde{\phi}(x_T)\|^2 = \langle \phi(x_T), \phi(x_T) \rangle + \langle \phi_0, \phi_0 \rangle - 2\langle \phi(x_T), \phi_0 \rangle \\ &= k(x_T, x_T) + \frac{1}{m^2} \sum_{i,j=1}^m \tilde{K}_{ij} - \frac{2}{m} \sum_{i=1}^m k(x_T, x_i) \end{aligned} . \tag{10}$$

The PC score x_T^{PCR} of the test point x_T in the reduced feature space F_R is equal to projection of its image $\tilde{\phi}(X)$ onto the eigenvectors V_R :

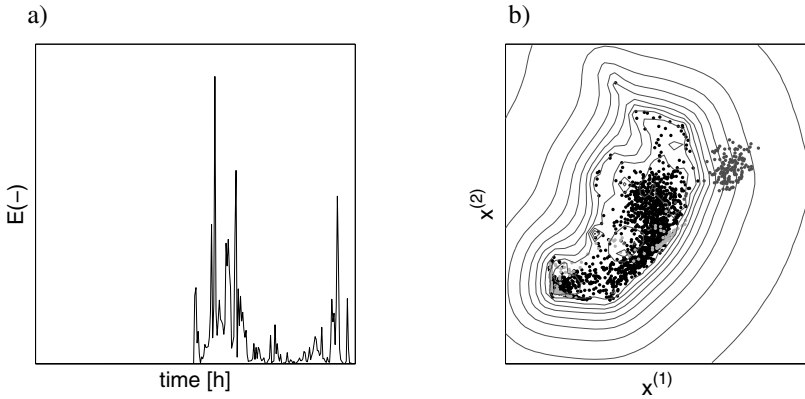


Fig. 2. a) Reconstruction error for a test set containing measurements from normal state (first half of the set) and leakage (second half of the set) b) The same test set in the input space : data points from the leakage (dark grey) can be separated from data points from normal operation (bright gray) by some chosen decision boundary (izolines). Model built from the training set (black).

$$x_T^{PCR} = \tilde{\phi}(x_T)V_R = \tilde{\phi}(x_T)\tilde{\phi}^T(X) \frac{U_R}{\sqrt{\Lambda_R}}, \tag{11}$$

where U_R and Λ_R contain s first eigenvectors and eigenvalues of \tilde{K} , respectively. The term $\tilde{\phi}(x_T)\tilde{\phi}^T(X)$ can be calculated using kernel function with correction that takes into account centring in the feature space resulting in vector K_T :

$$\begin{aligned} \tilde{\phi}(x_T)\tilde{\phi}^T(X) &= K_T = [K_{T1} \quad \dots \quad K_{Tr} \quad \dots \quad K_{Tm}] \\ K_{Tr} &= k(x_T, x_r) - \frac{1}{m} \sum_{i=1}^m \tilde{K}_{ri} - \sum_{i=1}^m k(x_T, x_i) + \frac{1}{m^2} \sum_{i,j=1}^m K_{ij}. \end{aligned} \tag{12}$$

Using (10) and combination of (11) and (12) the expression for reconstruction error $E(x_T)$ in (9) can be calculated. The value of the error is always between zero and a value close to one. An error $E(x_T)$ exceeding some selected maximal value of the reconstruction error E^{max} indicates that the test point x_T is not following the pattern defined by the training set X . This means that E^{max} serves as a decision boundary (Fig. 2) that enables to classify the current state of the system:

$$\begin{aligned} E(x_T) \geq E^{max} &\Rightarrow x_T \in FS \\ E(x_T) < E^{max} &\Rightarrow x_T \in NS \end{aligned} \tag{13}$$

4 Chojnice Case Study

In order to prove the efficiency of the presented method, a number of experiments has been carried out. Each single experiment aimed at answering the following question: “Is it possible to detect a leakage that occurred in node ‘A’ basing entirely on measurements taken at node ‘B’?”. All the experiments were based on the measurements provided by Epanet with simulations carried out using a calibrated model of a real network. This network has been divided into a number of regions that are partly independent in the sense of leakage discovery as described later. Training set corresponds to measurements collected during 6 days every 5 minutes, while the test set was collected during the 7th day, with the leakage being simulated in the middle of the day. For each of the monitoring nodes a KPCA model was built, with kernel width $\sigma=1$ and a size of the feature space F_R set to $s=70$. Values of these parameters were chosen as a result of an analysis. Since the data is normalized and the training set has the same size yielding the same dimension of the full feature space F , the result of applying common parameters for all nodes provides satisfactory outcome, however this leaves the place for further improvements. The third parameter maximal allowed reconstruction error E_{max} was chosen so that 99,9% of training set is considered to be in the normal state. In order to check what is the monitoring potential of each node, results from a set of following experiments were brought together: for a fixed monitored node leakages of a chosen size were consecutively simulated in adjoining nodes. This has provided a monitoring range map of a chosen node. Some nodes presents better performance than others. This is caused by the diverse effect of a certain leakage on the measurements. Simulations carried out proved that there are several factors that have strong influence on the performance of the node:

The position of monitored node in respect to the leakage node. As explained earlier a leakage causes disturbance within a limited range. This range is different for each node since compensation of adjoining nodes for water loss is never the same (Fig. 3). Furthermore, it turns out that the large supply pipes have the significant ability to mask the leakage symptom as they can easily deliver increased amount of water without affecting its own state. This motivates dividing the network into separate regions with the arrangement of supply pipes taken into account [7].

Size of the leakage. The influence of the leakage on the measured variables is in general proportional to the relative amount of water lost in a leakage. Larger leakages cause stronger disturbances and as a result larger area is affected. This explains why a monitoring node performs much better in case of larger leakages (Fig 3c).

The time of a day. The amount of the water flowing through the pipes changes throughout the day with the least water being supplied at night. Although the value of the pressure is generally lower at this time of the day resulting in less water being lost, it is much easier to observe the leakage as it has relatively stronger impact on the network. This means that the sensitivity of the monitoring node changes throughout the day.

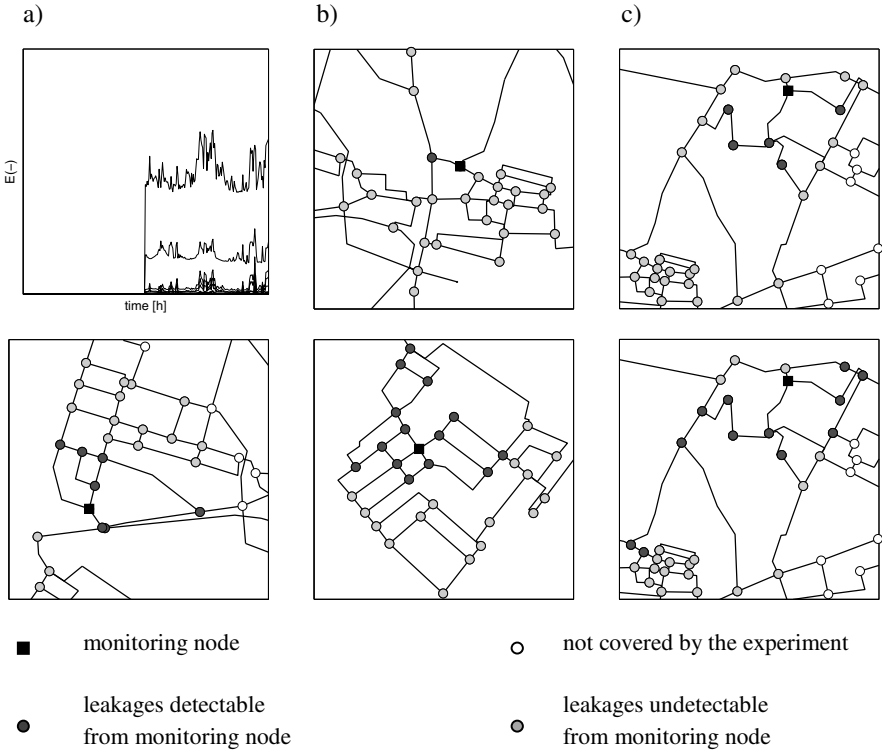


Fig. 3. a) (top) leakages simulated in different nodes have different impact on the reconstruction error $E(t)$ evaluated for monitored node, (bottom) results of the experiment presented on the map; b) poor (top) and good (bottom) candidate for monitoring node - simulated leakage $2 \text{ m}^3/\text{h}$, c) the range of detection depends on the size of the leakage: $1.5 \text{ m}^3/\text{h}$ (top), $12 \text{ m}^3/\text{h}$ (bottom)

Even though the area covered by the detection system differs for each monitoring node, they share some common properties: the area is always continuous and concentrated around monitoring node in a non-symmetric manner. The monitored nodes should be chosen carefully as there might be a significant difference in performance between a good candidate and a poor one (Fig. 3b). Setting a number of monitoring nodes provides a possibility to monitor an entire network. Since the area monitored by each node is subject to change depending on the leakage size, the number of required nodes heavily depends on the expected sensitivity of the system: if one needs to detect and to precisely localise even small leakages this requires setting a large number of monitoring nodes close to each other. The possibility to detect the leakages only in close neighbourhood of monitored node extends application of the method to localization of the potential fault (Fig. 4). If for a single node current value of reconstruction error $E(t)$ exceeds E^{\max} , it indicates that a leakage occurred in some close neighbourhood. If several nodes report an error at the same time this suggest that a larger leakage occurred somewhere in between.

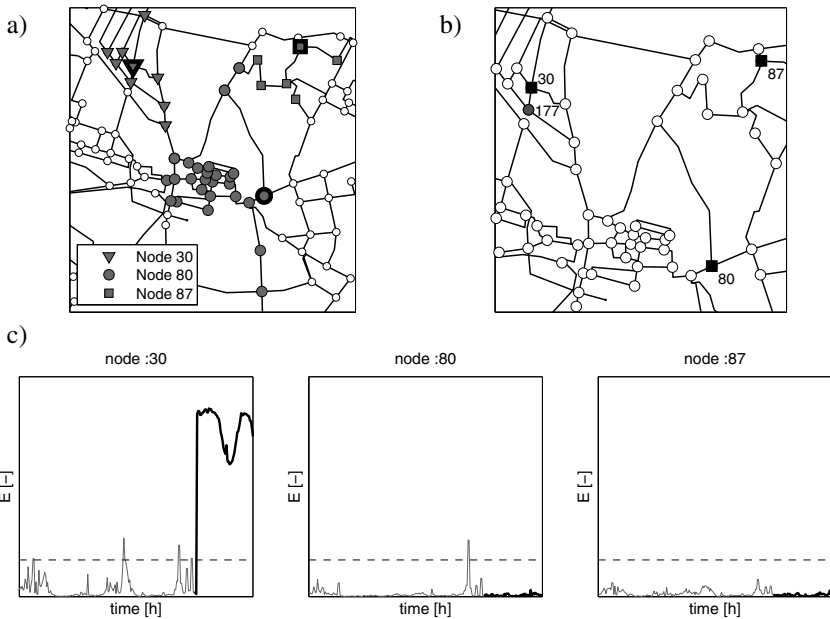


Fig. 4. The idea of leakage localisation using local models: a) an example of the detection range for three monitored nodes with simulated leakages $Q=2 \text{ m}^3/\text{h}$, b) monitored nodes marked with black squares, place of the simulated leakage marked in grey, c) values of reconstruction error in monitored nodes for situation given in b).

5 Conclusions and Future Work

The paper describes an approach to detect the leakages in water distribution system using Kernel PCA method with a limited number of measurements. The arrangement of the measuring points is determined through simulations and heuristics in order to ensure efficient fault detecting abilities of local KPCA models. By adjusting the number of controlled nodes, one can set a sensitivity of the system to maintain economic level of real losses. The usage of KPCA, instead of conventional PCA, reduces number of false alarms and prevents model conservatism. The methodology was verified on calibrated model and data of Chojnice town (Northern Poland). At this stage of the research localisation of the leakages is supervised by a man, however promising results completing the process of automatic fault detection and localisation are obtained by the paper Authors with usage of Self Organising Maps. Other faults (such as pump or valve breakdown, water contamination) can be identified and isolated using this approach. Moreover, the method is rather of a generic nature, hence might be transferred into similar systems e.g. pipeline systems, telecommunication systems, power systems etc, known as a network systems. Optimal and adaptive parameters of KPCA models selecting predispose the method to real time diagnostic and control systems e.g. Fault Tolerant Model Predictive Control.

References

1. Thornton, J., Sturm, R., Kunkel, G.: *Water Loss Control*. The McGraw-Hill Companies, New York (2008)
2. *Water Audits and Loss Control Programs - Manual of Water Supply Practices, M36*. American Water Works Association (2009)
3. Jin, Y., Yumei, W., Ping, L.: Leak Acoustic Detection in Water Distribution Pipeline. In: *The 7th World Congress on Intelligent Control and Automation*, pp. 3057–3061. IEEE Press, New York (2008)
4. Xiao-Li, C., Chao-Yuan, J., Si-Yuan, G.: Leakage monitoring and locating method of water supply pipe network. In: *The 7th International Conference on Machine Learning and Cybernetics*, pp. 3549–3551. IEEE Press, New York (2008)
5. Mashford, J., Silva, D.D., Marney, D., Burn, S.: An approach to leak detection in pipe networks using analysis of monitored pressure values by support vector machine. In: *3rd International Conference on Network and System Security*, pp. 534–539. IEEE Press, New York (2009)
6. Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.: A review of process fault detection and diagnosis. Part I, II, III. *Computers and Chemical Engineering* 27, 293–346 (2003)
7. Duzinkiewicz, K., Borowa, A., Mazur, K., Grochowski, M., Brdys, M.A., Jezior, K.: Leakage Detection and Localization in Drinking Water Distribution Networks by MultiRegional PCA. *Studies in Informatics and Control* 17(2), 135–152 (2008)
8. Jackson, J.E., Mudholkar, G.: Control procedures for residuals associated with principal component analysis. *Technometrics* 21, 341–349 (1979)
9. Schölkopf, B., Smola, A.J., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10, 1299–1319 (1998)
10. Aizerman, M., Braverman, E., Rozonoer, L.: Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control* 25, 821–837 (1964)
11. Hoffman, H.: Kernel PCA for novelty detection. *Pattern Recognition* 40, 863–874 (2007)

Decisional DNA Digital TV: Concept and Initial Experiment

Haoxi Zhang¹, Cesar Sanin¹, and Edward Szczerbicki²

¹Faculty of Engineering and Built Environment, School of Engineering,
The University of Newcastle, Callaghan, NSW, Australia 2308

²Gdansk University of Technology, Gdansk, Poland

haoxi.zhang@uon.edu.au,

Cesar.Sanin@newcastle.edu.au, Edward.Szczerbicki@zie.pg.gda.pl.

Abstract. Together with the booming popularity of Digital TV, the information of how viewers watch TV (preferences, content, viewing times, etc) could quite easily be available and extremely valuable. By running customized smart applications at viewers' set-top boxes, capturing and processing their viewing experiences, the TV program providers can not only collect information from their customers but also offer interactive contents to their viewers while improving customer service. In order to capture, reuse, and share the past experiences and preferences of how the viewers watch TV, we propose a novel application of the Decisional DNA to the Digital TV field. Decisional DNA is a domain-independent, flexible, smart knowledge representation structure which allows its domains to acquire, reuse, evolve and share knowledge in an easy and standard way. In this paper, we present the features, architecture and initial experimental results of our work.

Keywords: Decisional DNA, Set of Experience Knowledge Structure, Digital TV, XML.

1 Introduction

Nowadays, TV sets are experiencing a new revolution: as the progress of digitalization and computerization of our daily life, the TV set is becoming interactive, or even becoming a computer. There are many organizations and companies involved into implementation of interactive TV, like Google, SONY and Apple. Also, there have been a few solutions offered by them, such as Java TV and Multimedia Home Platform.

By using these existing solutions, developers can build interactive functions into Digital TV sets. In this paper, we introduce a domain-independent and standard approach, called the Decisional DNA Digital TV (DDNA DTV) to capture, reuse, and share viewers' TV watching experiences. It is based on the Java TV platform, and uses a novel knowledge representation structure – Decisional DNA.

This paper is organized as follows: section two describes an academic background on basic concepts related to our work; section three presents the features, architecture and experiments for the DDNA DTV Systems. Finally, in section four, concluding remarks are drawn.

2 Background

2.1 Digital TV

Digital television (DTV) is the television broadcasting system that uses the digital signals to transmit program contents. DTV not only delivers distortion-free audio and video signals; more importantly, it offers much higher radio spectrum efficiency than analog television does. DTV can also seamlessly integrate with other digital media, computer networks, and communication systems, enabling multimedia interactive services and data transmission [21].

A) Formats and Bandwidth

Digital television supports a range of different picture formats defined by the combination of interlacing, size, and aspect ratio (width to height ratio). With digital terrestrial television broadcasting in the world, the range of formats can be broadly divided into two categories: SDTV and HDTV. These terms by themselves are not very precise, and many subtle intermediate cases exist [18].

Standard definition TV (SDTV), by comparison, may use one of several different formats taking the form of various aspect ratios depending on the technology used in the country of broadcast. For 4:3 aspect-ratio broadcasts, the 640×480 format is used in NTSC countries, while 720×576 is used in PAL countries. For 16:9 broadcasts, the 704×480 format is used in NTSC countries, while 720×576 is used in PAL countries. However, broadcasters may choose to reduce these resolutions to save bandwidth (e.g., many DVB-T channels in the United Kingdom use a horizontal resolution of 544 or 704 pixels per line) [12].

High-definition television (HDTV), one of several different formats that can be transmitted over DTV, uses different formats, amongst which: 1280×720 pixels in progressive scan mode (abbreviated 720p) or 1920×1080 pixels in interlace mode (1080i). Each of these utilizes a 16:9 aspect ratio. (Some televisions are capable of receiving an HD resolution of 1920×1080 at a 60 Hz progressive scan frame rate — known as 1080p.) HDTV cannot be transmitted over current analog channels.

B) Standards

Currently, there are three main DTV standard groups [21]:

1) The Digital Video Broadcasting Project (DVB), a European based standards organization, which developed the DVB series of DTV standards, standardized by the European Telecommunication Standard Institute (ETSI) [9].

2) The Advanced Television Systems Committee (ATSC), a North America based DTV standards organization, which developed the ATSC terrestrial DTV series of standards. In addition, the North American digital cable TV standards now in use were developed separately, based on work done by Cable Television Laboratories (Cable Labs) and largely codified by the Society of Cable Telecommunications Engineers (SCTE) [2].

3) The Integrated Services Digital Broadcasting standards (ISDB), a series of DTV standards developed and standardized by the Association of Radio Industries and Business (ARIB) and by the Japan Cable Television Engineering Association (JCTEA) [1].

C) Reception

There are various ways to receive digital television. One of the oldest means of receiving DTV (and TV in general) is using an antenna (known as an aerial in some countries). This way is known as Digital Terrestrial Television (DTT) [19]. With DTT, viewers are limited to whatever channels the antenna picks up. Signal quality will also vary.

Other ways have been devised to receive digital television. Among the most familiar to people are digital cable and digital satellite. In some countries where transmissions of TV signals are normally achieved by microwaves, digital Multichannel Multipoint Distribution Service (MMDS)[11] is used. Other standards, such as Digital Multimedia Broadcasting (DMB) [20] and Digital Video Broadcasting - Handheld (DVB-H) [15], have been devised to allow handheld devices such as mobile phones to receive TV signals. Another way is Internet Protocol TV (IPTV) [3], which is receiving TV via Internet Protocol, relying on Digital Subscriber Line (DSL) or optical cable line.

Some signals carry encryption and specify use conditions (such as "may not be viewed on displays larger than 1 m in diagonal measure" or "may not be recorded") backed up with the force of law under the WIPO Copyright Treaty and national legislation implementing it, such as the U.S. Digital Millennium Copyright Act [17]. Access to encrypted channels can be controlled by a removable smart card, for example via the Common Interface (DVB-CI) standard for Europe and via Point Of Deployment (POD) for IS or named differently CableCard [9] [18].

2.2 Interactive Television

Interactive television (generally known as iTV) describes a number of techniques that allow viewers to interact with television content and services; it is an evolutionary integration of the Internet and DTV [10].

The most exciting thing of an interactive TV is the ability to run applications that have been downloaded as part of the broadcast stream: this is really what makes the difference between a basic digital TV box and an interactive TV system. In order to support and enable interactive applications, the receiver is required to support not only the implementation of APIs needed to run the applications, but also the infrastructure needed to inform the receiver what applications are available and how to run them.

Interactive TV has drawn attention from researchers, organizations, and companies, and there have been a few efforts and solutions offered by them. Java TV and Multimedia Home Platform are the two most popular and vibrant techniques in this field [14] [16].

A) Java TV

The Java TV is a Java-based software framework designed for supporting digital TV platforms from Sun Microsystems. It brings together a number of the common elements that are needed in a digital TV platform. These include the core application model and lifecycle, access to broadcast services (either via Java TV itself or via the Java Media Framework) and access to service information [14].

Most importantly, Java TV is not bound to a specific set of standards for digital TV. Java TV is explicit, pure, and independent. Because of this, it works equally well

with many solutions for digital TV, such as ATSC solutions, or OpenCable solutions, or DVB-based systems. It gives Java TV a very strong advantage that applications written to use Java TV APIs will work on any platform that supports it, rather than being tied to a specific broadcast system [14].

B) Multimedia Home Platform

Multimedia Home Platform (MHP) is an open standard middleware system designed by the DVB Project for enhanced and interactive digital television [16].

The MHP enables the reception and execution of interactive, Java-based applications on a TV-set. Interactive TV applications can be delivered over the broadcast channel, together with video and audio streams. These applications can be, for instance, games, e-mail, information services, interactive voting, shopping or SMS.

MHP defines a generic interface between interactive digital applications and the terminals, which those applications execute on. This interface decouples different applications of a provider from specific hardware and software details of different MHP terminal implementations. It enables digital content providers to address all types of terminals ranging from low-end to high-end set top boxes, integrated digital TV sets and multimedia PCs. The MHP extends the existing DVB open standards for broadcast and interactive services in various broadcasting networks, like satellite, cable or terrestrial networks.

2.3 Set of Experience Knowledge Structure (SOEKS) and Decisional DNA

The Set of Experience Knowledge Structure (SOEKS or shortly SOE) is a domain-independent, flexible and standard knowledge representation structure [13]. It has been developed to acquire and store formal decision events in an explicit way [4]. It is a model based upon available and existing knowledge, which must adapt to the decision event it is built from (i.e. it is a dynamic structure that depends on the information provided by a formal decision event) [8]; besides, it can be represented in XML or OWL as an ontology in order to make it transportable and shareable [5] [6].

SOEKS is composed of variables, functions, constraints and rules associated in a DNA shape permitting the integration of the Decisional DNA of an organization [8]. Variables normally implicate representing knowledge using an attribute-value language (i.e. by a vector of variables and values) [7], and they are the centre root of the structure and the starting point for the SOEKS. Functions represent relationships between a set of input variables and a dependent variable; moreover, functions can be applied for reasoning optimal states. Constraints are another way of associations among the variables. They are restrictions of the feasible solutions, limitations of possibilities in a decision event, and factors that restrict the performance of a system. Finally, rules are relationships between a consequence and a condition linked by the statements IF-THEN-ELSE. They are conditional relationships that control the universe of variables [8].

Additionally, SOEKS is designed similarly to DNA at some important features. First, the combination of the four components of the SOE gives uniqueness, just as the combination of four nucleotides of DNA does. Secondly, the elements of SOEKS are connected with each other in order to imitate a gene, and each SOE can be classified, and acts like a gene in DNA [8]. As the gene produces phenotypes, the

SOE brings values of decisions according to the combined elements. Then a decisional chromosome storing decisional “strategies” for a category is formed by a group of SOE of the same category. Finally, a diverse group of SOE chromosomes comprise what is called the Decisional DNA [4].

In short, as a domain-independent, flexible and standard knowledge representation structure, SOEKS and Decisional DNA provide an ideal approach which can not only be very easily applied to various embedded systems (domain-independent), but also enable standard knowledge communication and sharing among these embedded systems.

3 The Decisional DNA Digital TV

Nowadays, digital TV has being rolling on with full force. Thanks to its capability of transmitting digital data along with the audiovisual contents, the TV program providers can interact with their viewer by offering customized applications, which run at their viewers’ set-top boxes. In order to capture, reuse, and share viewers’ TV watching experiences, we applied the novel knowledge representation structure – Decisional DNA to digital TV, called The Decisional DNA Digital TV.

3.1 System Architecture

The DDNA DTV consists of the User Interface, the System I/O, the Integrator, the Prognoser, the XML Parser and the Decisional DNA Repository (see Fig. 1).

- **User Interface:** The User Interface is developed to interact with the user/viewer. In particular, user can control, set and configure the system by using the user interface. Like the user can use remote control to select services, give feedback to a movie and interact with the service provider through the User Interface.

- **System I/O:** The System I/O allows our Decisional DNA approach to communicate with its domain. The System I/O tells the DTV which service is selected, what movie should play, what feedback was given. Also, it reads the media stream, feedback, system time, service information from its domain.

- **Integrator:** In our case, we link each experience with a certain scenario. The Integrator is the place where the scenario data is gathered and organized, such as the system time, the name of a selected service, user input and other service information, describing the circumstance under which experience is acquired. Therefore, the scenario data is transformed into a set of experience, i.e. Variables, Functions, Constrains, and Rules. The Integrator organizes the scenario data into strings using ID□VALUE format, and send them to the Prognoser for further processing.

- **Prognoser:** The Prognoser is in charge of sorting, analyzing, organizing, creating and retrieving experience. It sorts data received from the Integrator, and then, it analyzes and organizes the data according to the system configuration. Finally, it interacts with the Decisional DNA Repository and the XML Parser in order to store and reuse experience depending on the purpose of different tasks.

- **XML Parser:** The XML Parser is the converter that translates knowledge statements generated by the Prognoser into the Decisional DNA experience structure represented in XML format; and interprets the retrieved XML-represented Decisional DNA experience for reusing.

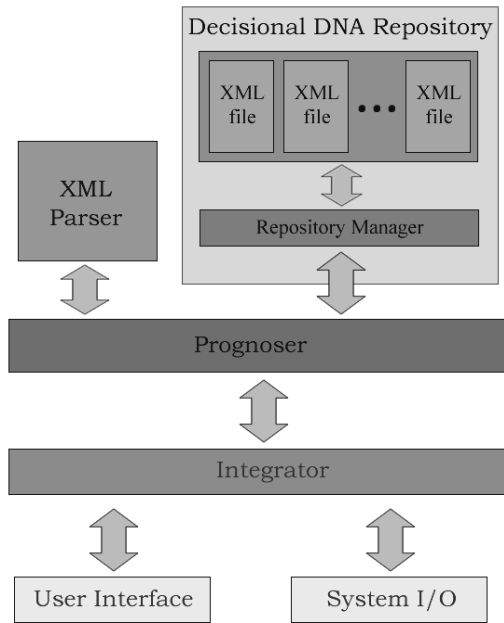


Fig. 1. System Architecture for DDNA DTV

- **Decisional DNA Repository:** The Decisional DNA Repository is the place where experiences are stored and managed. It uses standard XML to represent experiential knowledge, which makes standard knowledge sharing and communication become easier. It is composed of the Repository Manager and XML files:

- a) **Repository Manager.** The Repository Manager is the interface of the Decisional DNA Repository. It answers operation commands sent by the Prognoser and manages the XML files. There are two main tasks in it: searching experiences and managing XML files.

- b) **XML files.** We use a set of XML tags described in [13] to store Decisional DNA in XML files. In this way, Decisional DNA is explicitly represented, and ready to be shared among different systems.

3.2 Simulation and Experiments

Due to the lack of supporting from TV set companies and set-top-box manufactures, it is very hard to find a real TV set or set-top-box in which we can burn our software to. Thus, we used the Java TV SDK with NetBeans 6.8 on a DELL Latitude ES400 laptop to test the idea of the DDNA DTV.

At this stage, the main purpose of our experiments is to prove that the Decisional DNA can work with Java TV, and our approach can provide its domain with the ability of experience capturing and reusing.

We assume that there are some movies downloaded into user’s DDNA DTV, and we simulated a viewer watching movies on this DDNA DTV, Fig. 2 shows a screenshot of his TV. As we can see, the viewer’s screen is composed of four areas:

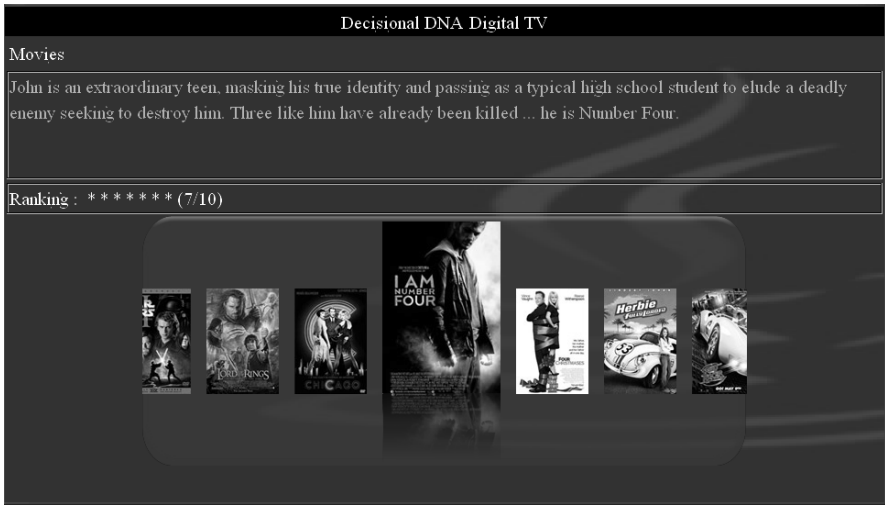


Fig. 2. Screenshot of DDNA DTV

Service Name which shows “Movies” here, Service Information which displays introduction of a selected movie here, Ranking, and Movie Showcase.

We capture viewer’s watching experience by recording the movie name, director, watch date, watch time, ranking, type, and viewer’s name. Once we have these seven variables, we can analyse viewer’s watching preference, and give the viewer better recommendations from his previous watching experience.

For example, we assume that there is a viewer, Tom, who likes to watch action movies on every Saturday night as shown in the Table 1.

In order to capture Tom’s movie watching experience, we record every movie he watched by storing seven variables: Name, Director, Watch Day, Watch Time, Ranking, Type, and Viewer. Name and Director are used to indicate which movie he watched. Watch Day and Watch Time tell which day and when he watched this movie. Ranking shows how he likes this movie. Type illustrates what kind of movie he watched. Viewer saves the name of user, in this case, it is Tom. Those variables are gathered and organized by the Integrator and then send to the Prognoser; finally, they are stored as a SOEKS in XML format [13] (See Fig. 3).

Table 1. Tom’s Movie Watching Records.

Movie Name	Watch Date	Watch Time	Ranking	Type
I am number four	8/01/2011	19:35	7	Action
Star Wars	22/01/2011	20:02	9	Action
Raiders of the Lost Ark	29/01/2011	20:13	8.5	Action
The Matrix	12/02/2011	19:42	8.7	Action
The Incredibles	19/02/2011	21:07	8.6	Action
The Mechanic	26/02/2011	21:19	7.5	Action

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!-- Set of Experience Knowledge Structure -->
- <set_of_experience xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <date>2011-01-08</date>
  <hour>19:35:41</hour>
- <category>
  <!-- Category encloses this SOE into a determined chromosome of the company -->
  <area><b>User Experience</b></area>
  <subarea><b>Watching TV</b></subarea>
  <subject><b>Movie</b></subject>
</category>
- <set_of_variables>
  <!-- Variables included in the model -->
- <variable>
  <var_name><b>Name</b></var_name>
  <var_type><b>CATEGORICAL</b></var_type>
  <var_cvalue><b>I AM NUMBER FOUR</b></var_cvalue>
  <var_evalue><b>I AM NUMBER FOUR</b></var_evalue>
  <unit />
  <internal><b>false</b></internal>
  <weight><b>0.0</b></weight>
  <l_range><b>0.0</b></l_range>
  <u_range><b>0.0</b></u_range>
- <categories>
  <category><b>I AM NUMBER FOUR</b></category>
</categories>
  <priority><b>0.0</b></priority>
</variable>
- <variable>
  <var_name><b>Director</b></var_name>
  <var_type><b>CATEGORICAL</b></var_type>
  <var_cvalue><b>D.J. CARUSO</b></var_cvalue>
  <var_evalue><b>D.J. CARUSO</b></var_evalue>
  <unit />
  <internal><b>false</b></internal>
  <weight><b>0.0</b></weight>
  <l_range><b>0.0</b></l_range>
  <u_range><b>0.0</b></u_range>
- <categories>
  <category><b>D.J. CARUSO</b></category>
</categories>
  <priority><b>0.0</b></priority>
</variable>
- <variable>
  <var_name><b>Watch Day</b></var_name>
  <var_type><b>CATEGORICAL</b></var_type>
  <var_cvalue><b>SATURDAY</b></var_cvalue>
  <var_evalue><b>SATURDAY</b></var_evalue>
  <unit />
  <internal><b>false</b></internal>
  <weight><b>0.0</b></weight>
  <l_range><b>0.0</b></l_range>
  <u_range><b>0.0</b></u_range>
- <categories>
  <category><b>SATURDAY</b></category>
</categories>
  <priority><b>0.0</b></priority>
</variable>
- <variable>
  <var_name><b>Watch Time</b></var_name>
  <var_type><b>CATEGORICAL</b></var_type>
  <var_cvalue><b>19:35:41</b></var_cvalue>
  <var_evalue><b>19:35:41</b></var_evalue>
  <unit />
  <internal><b>false</b></internal>
  <weight><b>0.0</b></weight>
  <l_range><b>0.0</b></l_range>
  <u_range><b>0.0</b></u_range>
- <categories>
  <category><b>19:35:41</b></category>
</categories>
  <priority><b>0.0</b></priority>
</variable>
- <variable>
  <var_name><b>Ranking</b></var_name>
  <var_type><b>NUMERICAL</b></var_type>
  <var_cvalue><b>7</b></var_cvalue>
  <var_evalue><b>7</b></var_evalue>
  <unit />
  <internal><b>false</b></internal>
  <weight><b>0.0</b></weight>
  <l_range><b>0.0</b></l_range>
  <u_range><b>0.0</b></u_range>
  <priority><b>0.0</b></priority>
</variable>
- <variable>
  <var_name><b>Type</b></var_name>
  <var_type><b>CATEGORICAL</b></var_type>
  <var_cvalue><b>ACTION</b></var_cvalue>
  <var_evalue><b>ACTION</b></var_evalue>
  <unit />
  <internal><b>false</b></internal>
  <weight><b>0.0</b></weight>
  <l_range><b>0.0</b></l_range>
  <u_range><b>0.0</b></u_range>
- <categories>
  <category><b>ACTION</b></category>
</categories>
  <priority><b>0.0</b></priority>
</variable>
- <variable>
  <var_name><b>Viewer</b></var_name>
  <var_type><b>CATEGORICAL</b></var_type>
  <var_cvalue><b>TOM</b></var_cvalue>
  <var_evalue><b>TOM</b></var_evalue>
  <unit />
  <internal><b>false</b></internal>
  <weight><b>0.0</b></weight>
  <l_range><b>0.0</b></l_range>
  <u_range><b>0.0</b></u_range>
- <categories>
  <category><b>TOM</b></category>
</categories>
  <priority><b>0.0</b></priority>
</variable>
</set_of_variables>
<set_of_functions />
<set_of_constraints />
<set_of_rules />
</set_of_experience>

```

Fig. 3. A SOEKS of Movie Watched

When the Prognoser recommends new movies to the user, it retrieves those stored watching experiences from the Decisional DNA Repository, and analyzes those experiences according to user’s settings. In this experiment, we analyze the movie types user watched, and what day in a week user usually watches them:

Each time when the user opens DDNA TV, the Prognoser retrieves all the user’s previous experiences. Then, the Prognoser analyzes what types of movie the user prefers on such a day. Finally, the Prognoser recommends new movies according to its analysis.

As we assumed, Tom usually watches action movies on every Saturday night; therefore, during a few weeks capturing experience, the system can learn and know his movie watching preference, i.e. Tom prefers action movies on Saturday nights. Thus, the system recommends him action movies on every Saturday night. Fig. 4 shows a screenshot of a newly recommended movie list for Tom.

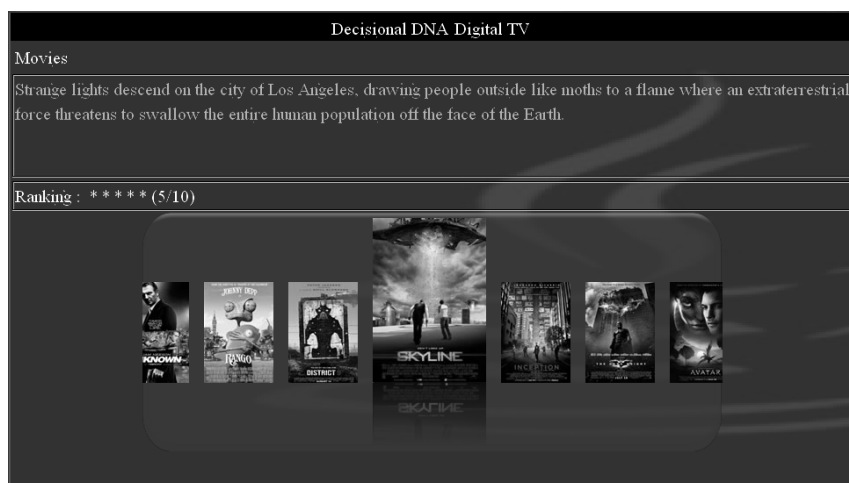


Fig. 4. Newly Recommended Movie List for Tom

4 Conclusions and Future Work

In this paper, we introduced the concept of the Decisional DNA Digital TV, and test it on a DELL laptop with Java TV SDK. As the result shows that the Decisional DNA can work with Java TV very well, and we can capture, store, and reuse viewers' TV watching experiences by using this approach, and provide TV viewers better user experience.

Since the DDNA DTV research is at its very early stages, there are quit a few further research and refinement remaining to be done, some of them are:

- Refinement of the requirements and system design of DDNA DTV.
- Enhancement of the efficiency of Decisional DNA Repository storage and query.
- Further development of the user login system.
- Refinement and further development of algorithm using in the Prognoser.
- Literature review of interactive TV, and find a proper way to adapt this idea into the real DTV environment.

References

1. ARIB, the Association of Radio Industries and Business, <http://www.arib.or.jp/english/>
2. ATSC, the Advanced Television Systems Committee, <http://www.atsc.org/cms/>
3. Yarali, A., Cherry, A.: Internet Protocol Television (IPTV). TENCON 2005 IEEE Region 10, 1–6 (2005) ISBN: 0-7803-9311-2
4. Sanin, C., Szczerbicki, E.: Experience-based Knowledge Representation SOEKS. Cybernetics and Systems 40(2), 99–122 (2009)
5. Sanin, C., Szczerbicki, E.: Extending Set of Experience Knowledge Structure into a Transportable Language Extensible Markup Language. International Journal of Cybernetics and Systems 37(2-3), 97–117 (2006)

6. Sanin, C., Szczerbicki, E.: An OWL ontology of Set of Experience Knowledge Structure. *Journal of Universal Computer Science* 13(2), 209–223 (2007)
7. Lloyd, J.W.: *Logic for Learning: Learning Comprehensible Theories from Structure Data*. Springer, Berlin (2003)
8. Sanin, C., Mancilla-Amaya, L., Szczerbicki, E., CayfordHowell, P.: Application of a Multi-domain Knowledge Structure: The Decisional DNA. In: *Intel. Sys. For Know. Management. SCI*, vol. 252, pp. 65–86 (2009)
9. DVB - The Digital Video Broadcasting Project, <http://www.dvb.org/>
10. Schwalb, E.: *iTV Handbook: Technologies & Standards*. *ACM Computers in Entertainment* 2(2), Article 7 (2004) ISBN: 0131003127
11. IEEE, “Multichannel Multipoint Distribution Service”, <http://grouper.ieee.org/groups/802/16/>
12. Latest snapshots - Freeview/DTT bitrates, “Latest snapshots”, <http://dtf.me.uk/>
13. Maldonado Sanin, C.A.: *Smart Knowledge Management System*, PhD Thesis, Faculty of Engineering and Built Environment - School of Mechanical Engineering, University of Newcastle, E. Szczerbicki, Doctor of Philosophy Degree, Newcastle (2007)
14. TV Without Borders, “Java TV Tutorial”, <http://www.interactivetvweb.org/tutorials/javatv/>
15. Reimers, U.H.: DVB-The Family of International standards for Digital Video broadcasting. *Proceedings of the IEEE* 94(1), 173–182 (2006) ISSN: 0018-9219
16. Vrba, V., Cvrk, L., Sykora, M.: Framework for digital TV applications. In: *Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning*, p. 184 (2006) ISBN: 0-7695-2552-0
17. Wikipedia, “Digital Millennium Copyright Act”, http://en.wikipedia.org/wiki/Digital_Millennium_Copyright_Act
18. Wikipedia, “Digital Television”, http://en.wikipedia.org/wiki/Digital_television/
19. Wikipedia, “Digital Terrestrial Television”, http://en.wikipedia.org/wiki/Digital_terrestrial_television
20. World DMB, Digital Multimedia Broadcasting, <http://www.worlddab.org/>
21. Wu, Y., Hirakawa, S., Reimers, U., Whitaker, J.: Overview of digital television development worldwide. *Proc. IEEE* 94(1), 8–21 (2006)

Application of Program Agents for Optimisation of VoIP Communication

Hrvoje Očevčić¹ and Drago Žagar²

¹ Hypo Alpe-Adria-Bank d.d., Zagreb, Croatia

² University of Osijek, Faculty of Electrical Engineering

hrvoje.ocevcic@hypo-alpe-adria.hr, drago.zagar@etfos.hr

Abstract. In this paper the model for simple management and optimisation of VoIP communication quality is proposed. The measurement values are obtained from operative network, as well as under the same conditions, the subjective information on the quality of service. One of the paper objectives is to compare the experimental results of MOS values with calculated values of parameter R. Therefore, the measured values are compared and incorporated in adjusted E-model. This simplified model is presented in framework of constructing the model for VoIP communication optimisation based on simple measurement information as a base for the proposal of agent architecture for optimisation of VoIP communication quality.

Keywords: VoIP, Quality of Service, delay, Mean Opinion Score (MOS), R factor, Agents.

1 Introduction

E-model is defined by ITU-T association in order to assess the quality of speech and connection of subjective and objective communication parameters. The output values of E-mode are scalar, and they are called R-values or R-factor. R factor can serve for calculation of subjective Mean Opinion Score (MOS) value [9]. It is possible to adjust E-model and with certain limitations, to apply it in VoIP systems [8].

The paper suggests the adjusted E-model where R factor and the corresponding R values are described. According to ITU-T standard G.107 [8], the basic values of the majority of parameters are defined with the same network conditions. The use of pre-defined values of parameters in E-model gives R factor of 93.2.

The experiment was carried out consisting of two parts, calculation of MOS value with the adjusted model and measuring the subjective measurement values with listeners' participation. Regarding the constant terms in which the experiment was carried out, the adjustment of E-model is applicable. The adjustment of E-model has been elaborated through many papers and systems based on research of VoIP networks, [9].

The result of the paper is the correlation between the objective and subjective physical measuring in view of constructing the proactive system for optimisation of the quality of service in VoIP communication.

2 Theoretical Premises

The fields that can be used to influence the quality of service in the wholesome network can be divided to two sections. The first section pertains to the software support installed at the end computer, for instance, operational system. The second section pertains to the very network that transfers data from and to the end computers. This paper focuses mainly on the second section.

2.1 The Parameters of the Quality of Service with Transfer of Voice over IP Networks

In order to make the transfer of voice service over IP network a quality replacement for the standard phone service of voice transfer over PSTN network, the users should be provided at least the same speech quality. As other services of transfer in real time, IP telephony is very sensitive to delay and oscillation in delay. The techniques for providing the quality of service make it possible that voice packages have special treatment, necessary for achievement of the desired quality of service [3], [4]. QoS techniques imply the following procedures:

- supporting the allocated bandwidth,
- reduction of package loss,
- avoiding and managing network congestion,
- shaping network traffic,
- use of strategies of traffic priorities in the network.

2.2 Adjusted E-Model for VoIP Communication

In many VoIP networks the implementation of E-model can be questionable as it is very often hard to find the proper connection between VoIP and classic telephony. The example could be dialling the number by the user in classic telephony that is re-directed at the central office, coded and further processed as VoIP. Here many elements required for calculation in E-model are not measurable and therefore it is necessary to adjust the model by implementation of some new values and assumptions. There are many papers [6] considering adjustment of the current ITU-T E-model for application in characteristic environments.

In case the conditions for deployment of conversation between the two speakers are the same, the ITU-T P.800 standard suggests the application of basic, implied values for the majority of parameters. If the basic values are listed for all other parameters, R factor becomes dependent on only one parameter - delay and thus the delay is the parameter to compare the measuring results. The adjustment of E-model is presented below:

R factor is presented as:

$$R = R_o - I_s - I_d - I_e + A \quad (1)$$

Where R_o is the basic signal-noise ratio, includes the noise sources as are the *circuit noise* and the *room noise*,

$$R_o = 15 - 1.5(SLR + N_o) \tag{2}$$

The further elaboration of equations gives the form to which it is possible to list the basic values [8]. By elaboration of E-model with use of values stated in Annex A6, the basic signal-noise ratio is:

$$R_o = 94.77.$$

The sum of all impairments, I_s can be shown as [8]:

$$I_s = I_{olr} + I_{st} + I_q \tag{3}$$

And after the listing of basic values it is

$$I_s = 1.43.$$

Delay impairment, I_d , is defined with three worsening factors:

$$I_d = I_{dte} + I_{dle} + I_{dd} \tag{4}$$

Using the basic values and assumptions based on ITU-T standards, we get:

$$I_d = 0,15 + I_{dd}$$

And ultimately, R parameter is:

$$\begin{aligned} R &= R_o - I_s - I_d - I_e + A = \\ &= 94,77 - 1,43 - (0,15 + I_{dd}) - 0 + 0 = \\ &= 93,19 - I_{dd} \end{aligned} \tag{5}$$

In what I_{dd} depends on one variable parameter of these measuring, that is, on delay T_a , as already mentioned. For the delay of less than 100ms, it holds true $I_{dd} = 0$ [9] (i.e. basic value), and finally [1]:

$$R = 93,19.$$

2.3 Connecting R Factor and MOS Value

The importance of R-factor is in that it can be calculated in real time with measuring on specific transfer line and communication equipment. With correct modelling, the link can be defined between R factor and subjective MOS value with high accuracy.

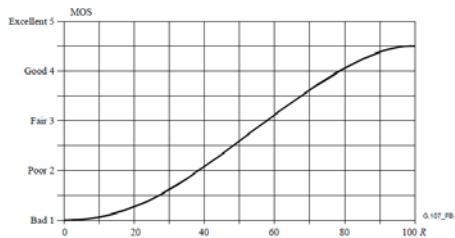


Fig. 1. The link between MOS value and R factor

Fig. 1 shows the functional dependence of MOS value with R factor. E-model defines the ideal case when $R=100$, and after defining the degrading values it is

deducted from number 100. In that way R factor is obtained on the specific link. Fig. 1 shows the relation between the user’s satisfaction with the service and the corresponding MOS value or R value.

R value can be linked to MOS value with mathematical expression:

$$MOS = \begin{cases} 1; & R < 0 \\ 1 + 0,035R + R(R - 60)(100 - R) * 7 * 10^{-6}; & 0 < R < 100 \\ 4,5; & R \geq 100 \end{cases} \quad (6)$$

MOS value 1 is defined for situations where R could be mathematically calculated to value less than zero. The standard also states the formula for the calculation of R value from the known MOS value. [8].

3 Experimental Model and Test Settings

Test setting is shown in Fig. 2. Between these two locations there is a leased line with bandwidth of 2Mbps over which the communication is established. At the both sides of the link there are access routers whose configurations are not changed.

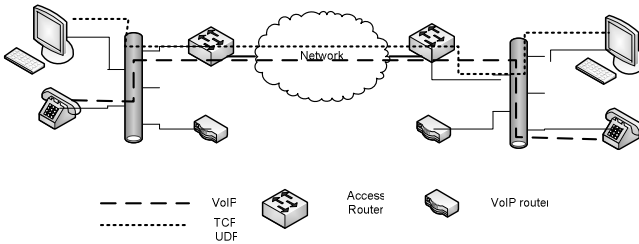


Fig. 2. Measuring test settings

For each of individual queuing, individual link bandwidth and individual codecs, the measuring process can be described in the following way:

The measuring is executed in a 100 seconds interval starting with VoIP traffic. After the first 30 seconds, TCP traffic is added, and after the following 30 seconds the additional UDP traffic is added. TCP and UDP traffics are generated synthetically by the software. It is important to emphasize that this UDP traffic is not VoIP traffic, and thus its parameters were not considered in the result analysis. The settings of VoIP system must be identical with different measuring in order to compare them and in order to confirm the thesis of comparison of objective and subjective approach.

3.1 Measuring Objective Parameters of Quality of Service on Experimental Model

The obtained data can be considered as objective indicators because they depend exclusively on the applied device, or equipment, and not on the speakers and their subjective opinion. The goal is to assess the quality felt by the receiver, after the data

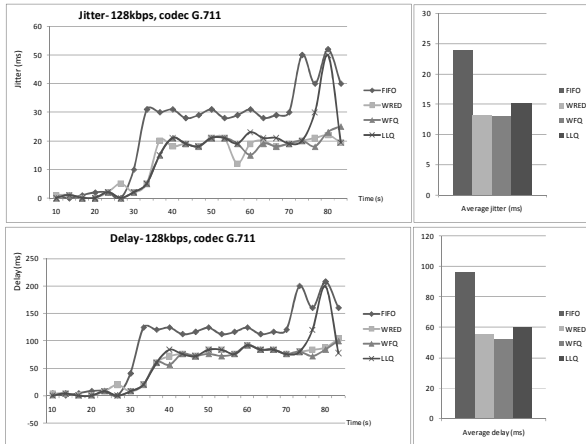


Fig. 3. Delay, delay oscillations and the mean values at 128kbps and G.711

have passed through the whole way over the network from the sender, the packages of sender are analysed as caught at the receiver's side.

The example of obtained measuring are shown by charts (Fig. 3) presenting the delays and jitter as well as the difference between mean values for G.711 voice codec with bandwidth of 128kbps. The best results were obtained by usage of codec G.711 in combination with WFQ queuing.

3.2 Measuring MOS Values and Subjective Assessment of VoIP Quality

Subjective assessment of speech quality was carried out by the model of Mean Opinion Score based on assessment by 14 interlocutors. The test speech sequence was transferred by network and "listened to" at the other side, lasting for 100 seconds, similar as the previous measurements. All other link parameters were identical as in the previous measuring (bandwidth, codecs, traffic types and queuing). The text was reproduced from digital record, according to appendix A4.4, of the standard [9]. For the comparison to the first part of measuring providing objective assessments, the sound record was added TCP (the second record) and UDP traffic (the third record).

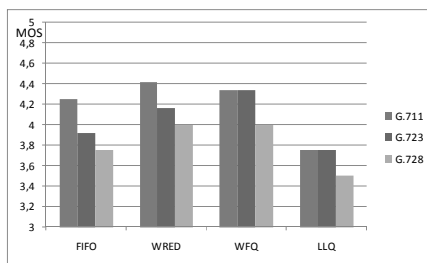


Fig. 4. The assessment of quality of the record listened to at 128kb/s

Fig. 4 shows the results of subjective measuring for the same scenario as measuring presented by Fig. 3. The best results were obtained by usage of codec G.711 in combination with WFQ queuing.

3.3 Comparison of Subjective and Objective VoIP Quality Assessment

The measuring on the experimental model provided the values used for calculation of the corresponding MOS values. Under the same conditions the measuring on the network was carried out according to ITU-T P.800 [9]. Here the MOS values were provided from the listeners according to the standard terms. These values were compared in order to prove the regularity of the paper thesis.

The comparison indicates the largest difference after the first phase of listening where the listeners followed the sound record on the network with VoIP traffic only. After that phase, for half a time the mixed traffic of VoIP and UDP was flowing through the network, and at the end also TCP was added. The listeners assessed the quality with MOS values at the end of record when the influence of the highest congestion was the strongest. Fig. 5 shows the differences of the calculated and measured MOS values with WRED queuing mechanism.

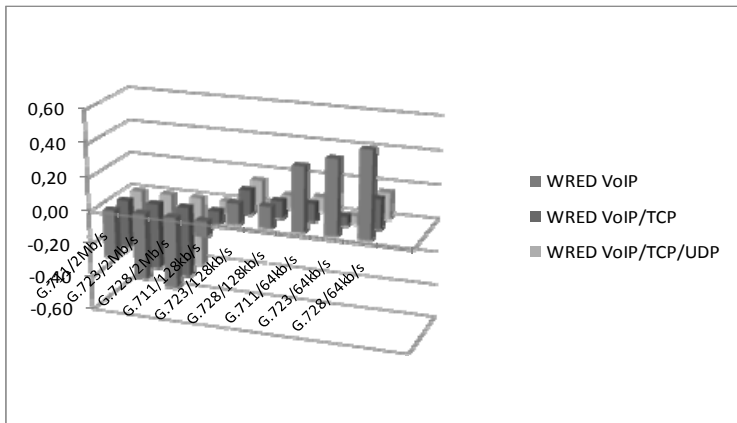


Fig. 5. Differences of MOS values with WRED servicing

The calculation of subjective MOS values for VoIP communication could be the base for adequate replacement of expensive and complex measuring and equipment, as well as for dynamic VoIP system optimisation.

4 Agent Architecture Proposal for VoIP System Optimisation

This paper proposes the method for VoIP communication optimisation by managing of service quality through adjusted E-model. The reasons for selection of agent architecture and service quality management approach are the following:

- Simplicity of implementation: it is possible to use software agents new hardware implementation is not necessary,

- Adjustability and flexibility: the work of agents does not depend on the type of environment, used devices and software, respectively,
- Surveillance: it is possible to supervise and maintain the agent system easily and effectively,
- Price: the costs of implementation of such system are reasonable, and the benefits from the system that can provide information on service quality in a simple way are significant.

Subjective approach to measuring also has advantage in relation to objective measuring. The following reasons can be stated as the major reasons for selection of subjective measuring:

- Clarity and simplicity of service quality gradation at all communication sides, with providers but also with users,
- Compliance of results with objective measuring requiring more engagement, equipment and traffic,
- Optimized system for service quality support is a great objective in communication service providing; it is also substantial to achieve cost benefit.

There are many commercial solutions [7], offering systems for continuous measuring of quality of service parameters of different Internet services. Commercial solutions are usually expensive and often require significant system implementations. Monitoring itself is often the cause of delay as well as specific noise in network thus the goal is to come up with simple surveillance mode of network parameters and service quality management.

4.1 Example of Optimising Parameters of VoIP Communication

The analysis of system represents the basis for VoIP communication service quality management and based on the analysis the optimisation model of VoIP communication is suggested on the grounds of agent architecture. The proposal of network architecture for optimisation of VoIP communication is shown in Fig. 6 where the topology similar to Call Admission Control system can be found where agents are managed by central unit.

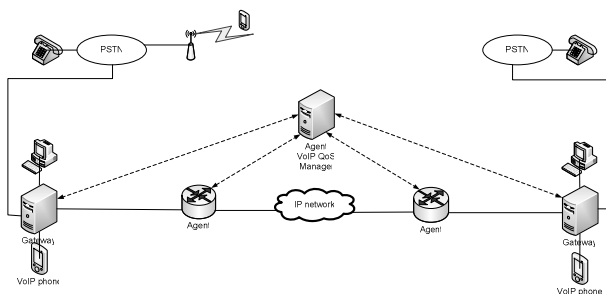


Fig. 6. Proposal of agent based architecture

The agents serve for communication and measuring of delay, and they continuously send the information on measuring values to VoIP QoS Manager which manages parameters of the session according to the defined guidelines. Also, the information is exchanged about the traffic type, bandwidth, codec, type of servicing and clients in session.

The guidelines can be initially defined or can be based on measuring and conclusions of this paper and good practices of VoIP communication. VoIP QoS Manager, should communicate also with other peripheral parts of the system, as presented concisely in the figure, but actually those are the parts of VoIP system with developed communication and work rules.

Measuring agents can send the information on need for interventions in communication channel at the right time and also deliver the relevant measuring information. Many of these information already make the component of the existent communication in VoIP systems. Fig. 7 shows the optimisation circle of VoIP communication with agents' mode.

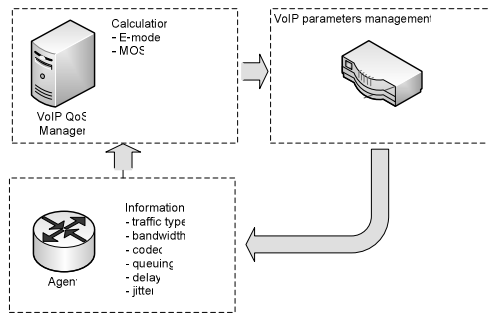


Fig. 7. Optimisation of VoIP communication

According to Fig. 6 VoIP session can be established with different scenarios and between the clients of different types. Optimising and managing parameters of VoIP communication must include also the optimisation of these non VoIP sessions. Thus it is very important to define the guidelines setting the work and behaviour of such system, therefore monitoring of delay values is the easiest way to regulate different communication circles.

The agents mutually exchange information on network parameters, and the analysis found the delay to be the most important and therefore, it should be measured and optimised. The result of experiments can be shown also in the form of the table (Table 1) stating optimal values of parameters for each tested scenario.

The advantage of such way of managing VoIP service quality is in that the various scenarios of network environments, reduced to delay measuring with existent monitoring of the basic network settings. With construction of such architecture the data on type of link used by one side is irrelevant (Mobile, PSTN, WAN, LAN ...) because on each session the same parameter is measured and the quality of service of VoIP communication is optimised in line with its amount.

Table 1. Optimal codec combinations

<i>Queing</i>	<i>Bandwidth (kb/s)</i>	<i>VoIP/TCP/UDP</i>	<i>Optimal codec</i>
FIFO	128	VT	G.722
FIFO	128	VTU	G.723
FIFO	64	VT	G.728
FIFO	64	VTU	G.728
WRED	128	VT	G.711
WRED	128	VTU	G.711
WRED	64	VT	G.711
WRED	64	VTU	G.711
WFQ	128	VT	G.711
WFQ	128	VTU	G.711
WFQ	64	VT	G.728
WFQ	64	VTU	G.711
LLQ	128	VT	G.711
LLQ	128	VTU	G.723
LLQ	64	VT	G.711
LLQ	64	VTU	G.723 / G.728

The pseudo-code is presented further whose copy to technological solution defines the work of agents which are the result of this research.

Parameters:

PS=bandwidth
 RP=queue
 C=codec
 VP=traffic_type (V=VoIP,
 VTU=VoIP+TCP+UDP)
 K=delay

'Before session:

If VP=VTU or PS<128 then
 Set VP=V 'allow only VoIP traffic
 Else
 Set RP=LLQ 'best score is LLQ
 Endif

'Agent loop:

If K>150ms then
 If PS=64 then
 Set VP=VoIP 'prioritization of traffic
 Endif
 Else
 If PS<128 then
 Use cRTP 'use header compression
 Set C=G.723
 Optimize_packet_size 'packet size optimization
 Jitter_buffer ' jitter buffer optimization
 Else
 Set VP=VoIP 'traffic prioritization
 Endif
 Endif

The described system does not imply the implementation of new device or software support; it is possible to implement it with minimal investment into the systems containing basic elements of VoIP communication. The system is adaptive and it is possible to implement it with numerous combinations of network elements because for managing VoIP quality of service it uses simple measuring and basic calculations.

5 Conclusion

There are many ways to manage the quality of service of VoIP communication, and here the agents' mode of managing is presented in which optimal parameters are chosen that can be economically managed and the service of quality received by the final user can be optimised. As simplicity and minimal software and hardware changes in the system were the prerequisite for economic and effective management of the quality of service, the paper presents the system for optimisation of VoIP communication with minimal changes and software upgrades in the current system.

The model of service quality analysis presented in this paper and the associated analyses can be applied as general solution to real time applications, for different measuring scenario, type of equipment, signalisation protocol, and service quality optimisation techniques.

The most important condition for successful optimisation found in analysis of VoIP quality of service is monitoring the delay that should be below the defined limit. There are numerous ways to influence the quality of service with VoIP communication and the centralised way of management selected in this paper has shown the advantages and benefits of use.

References

1. Black, U.: *Internet Telephony: Call Processing Protocols*. Prentice Hall, Englewood Cliffs (2001)
2. Rajan, R., Verma, D., Kamat, S., Felstaine, E., Herzog, S.: *A Policy Framework for Integrated and Differentiated Services in the Internet*. IEEE Network (September/October 1999)
3. *Understanding the Basic Networking Functions, Components, and Signaling Protocols in VoIP Networks*, Part Number: 200087-002 (2007)
4. Ismail, M.N.: *Analyzing of MOS and Codec Selection for Voice over IP Technology*, Annals. Computer Science Series. 7th Tome 1st Fasc. (2009)
5. Meddahi, A., Afifi, H.: *Packet-E-Model: E-Model for VoIP quality evaluation*. Computer Networks 50, 2659–2675 (2006)
6. *Understanding the Basic Networking Functions, Components, and Signaling Protocols in VoIP Networks*, Part Number: 200087-002 (May 2007)
7. *The E-model, a computational model for use in transmission planning*, ITU-T G.107, 1998, rev. (2000)
8. *Methods for subjective determination of transmission quality*, ITU-T P.800 (1996)

Study of Diabetes Mellitus (DM) with Ophthalmic Complication Using Association Rules of Data Mining Technique

Pornnapas Kasemthaweesab and Weresak Kurutach

Faculty of Information Science and Technology,
Mahanakorn University of Technology, Bangkok, 10530, Thailand
{u9910001, weresak}@mut.ac.th

Abstract. Association Rule Discovery is a significant data mining technique. In this paper, we applied this technique to discover fundamental association among a data set of diabetes mellitus (DM) patients with ophthalmic complication using a classifier based on gender, age and payment method of treatment expense. The result indicated that “diabetes mellitus (DM) patients Type II aging between 60-69 years old with no occupation whose payment for their treatment expense was by Government Official Rights of Continuous Treatment tended to have diabetes mellitus (DM) with ophthalmic complication.” This conclusion is useful for healthcare treatment of adulthood patients, welfare improvement of public healthcare, provision of helpful recommendation for diabetes mellitus patients and further development in finding disease complication.

Keywords: Data mining, Association Rule, Diabetes Mellitus.

1 Introduction

Data mining is a significant process of discovering knowledgeable information. [1], [2]. It is a part of knowledge management methods (Searching, Collecting, Dissemination and Applying) which aim to analyze sets of information or mass database required for seeking a distinctive pattern of relationships.

In Thailand, patients suffering from diabetes mellitus face with multiple complications, for example, neurological, ophthalmic, peripheral circulatory and renal complications resulting in high expense for treatment or permanent physical handicap due to certain complications. The discovery of relationship among groups of age, occupation and gender to find out a tendency of diabetes mellitus complications is a primary preventive measure to help diabetes mellitus patients for their better behavioural treatment.

There are some medical researches currently in diabetes mellitus using a variety of data mining techniques. [3], [4], [5], [6], [7] Those techniques can be categorized into 2 major purposes: one for prediction and the other for explanation purposes. In this study, data mining technique is for the latter purpose. The method focuses on searching information for noticeable patterns which are related to one another.

The study presented a technique of discovering association rules of diabetes mellitus (DM) with ophthalmic complication or linkage among groups of data. This data mining method looked for every relationship of interest from database derived from diabetes mellitus patient records in 2010. 65,535 raw data samples were normalized to 29,823 feasible profiles before bringing to solve for association with ophthalmic complication. By applying machine learning process, the data were classified and clustered to 3,964 patients with ophthalmic complication. Apriori algorithm method was a tool to analyze patients’ profile such as gender, age, occupation and methods of expense for medical treatment. The result brought about healthcare improvement and led to further medical research and development.

2 Theory and Research Involved

2.1 Data Mining

Data mining [8] is a process of extracting desired information from large data sets. It is an important tool to predict a tendency and behavior based on previous statistical inferences and transform data into useful intelligence giving an information advantage. A definition of data mining means user extracts information by using a process of data evaluation and verification in details. This process includes studying of past and present data. The ultimate goal is to derive useful and valid information from unknown data. Valid information is actionable and beneficial for modern business in decision making. Data mining is currently deployed in wide range of profiling practices such as marketing, scientific discovery and healthcare improvement. Data mining technique is a method of knowledge discovery in database (KDD) by sampling and analyzing portions of the larger population data set such as an analysis of consumer behaviors, weather forecast. The overall process is shown in Fig 1.

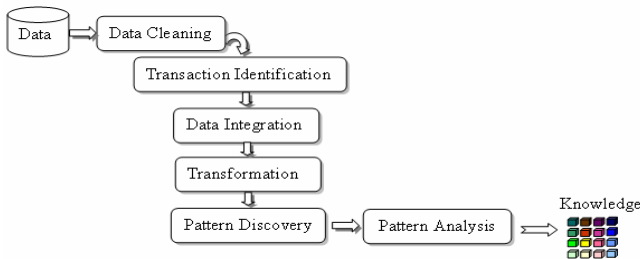


Fig. 1. Data Mining Process

2.2 Association Rules

Association rule mining finds interesting associations and/or correlation relationships among large set of data items. Association rules show attributed value conditions that occur frequently together in a given dataset. Association rules provide information of this type in the form of “if-then” statements. These rules are computed from the data and, unlike the if-then rules of logic, association rules are probabilistic in nature.

Association rules can be written in an algorithmic form of $X \rightarrow Y$ while $X, Y \subseteq I$ and $X \cap Y = \emptyset$ $X \rightarrow Y$ [Support = S , Confidence = C] means $X \rightarrow Y$ has S as a member in Transactions set of Database D . $S\%$ and $C\%$ of total transactions consists of item sets X and Y ($X \cup Y$). Association rules with Support ($X \cup Y$) not less than Minimum Support and Confidence ($X \rightarrow Y$) not less than Minimum Confidence are considered to be an interested rule.

2.3 Diabetes Mellitus (DM)

Diabetes Mellitus (DM) is a group of metabolic diseases in which a person has high blood sugar, either because the body does not produce enough insulin, or because cells do not respond to the insulin that is produced. This high blood sugar produces the classical symptoms of polyuria (frequent urination), polydipsia (increased thirst) and polyphagia (increased hunger). Diabetes mellitus, commonly referred to as diabetes, was first identified as a disease associated with "sweet urine." Over time, diabetes can lead to blindness, kidney failure, and nerve damage. These types of damage are the result of damage to small vessels, referred to as micro vascular disease.

There are roles for patient education, dietetic support, sensible exercise, with the goal of keeping both short-term and long-term blood glucose levels within acceptable bounds. In addition, given the associated higher risks of cardiovascular disease, lifestyle changes are recommended to control blood pressure.

2.4 Diabetes Mellitus (DM) with Ophthalmic Complication

The eyes can be affected in several ways by diabetes mellitus. The tiny blood vessels that sense light at the back of the eye are damaged, leading to blurred vision, sudden blindness, or black spots, lines, or flashing lights in the field of vision. The disorder occurs most frequently in patients with long-standing poorly controlled diabetes mellitus. Repeated hemorrhage may cause permanent opacity of the vitreous humor, and blindness may eventually result.

Diabetic retinopathy is the result of microvascular retinal changes. Hyperglycemia-induced intramural pericyte death and thickening of the basement membrane lead to incompetence of the vascular walls. These damages change the formation of the blood-retinal barrier and also make the retinal blood vessels become more permeable. Small blood vessels- such as those in the eye-are especially vulnerable to poor blood sugar (blood glucose) control [9].

Severe nonproliferative diabetic retinopathy enters an advanced, or proliferative, stage when blood vessels proliferate [10] (ie grow). The lack of oxygen in the retina causes fragile, new, blood vessels to grow along the retina and in the clear, gel-like vitreous humour that fills the inside of the eye. Severe nonproliferative diabetic retinopathy enters an advanced, or proliferative, stage when blood vessels proliferate. The lack of oxygen in the retina causes fragile, new, blood vessels to grow along the retina and in the clear, gel-like vitreous humour that fills the inside of the eye. The

Table 2. Types of complications in diabetes mellitus (DM) patients

<i>ID_Type</i>	<i>Diabetes Mellitus (DM) with Complication</i>
E10	Insulin-dependent diabetes mellitus
E100	Insulin-dependent diabetes mellitus type 1 with coma
E101	Insulin-dependent diabetes mellitus type 1 with ketoacidosis
E102	Insulin-dependent diabetes mellitus type 1 with renal complications
E103	Insulin-dependent diabetes mellitus type 1 with ophthalmic complications
:	:

Table 3. Exemple Occupational Data

<i>ID_OCCU PATION</i>	<i>Occupation name</i>	<i>ID_OCCU PATION</i>	<i>Occupation name</i>
00	Retired Government Official	63	Machine Driver
01	Hireling	71	Tailor
02	Farmer	72	Shoemaker
03	Government Official	88	Labor
04	State Enterprises	90	Police, Fireman
05	Trade	91	Servant, Cook
06	Government Sector	92	Waiter, Waitress
07	Student	94	Beautician
08	Housework	95	Laundry
09	Priest	A3	Doctor
11	Manager, Director	A4	Nurse
12	Reporter	A6	Teacher
13	Businessman	A7	Lawyer
15	Accountant	unknown	No information
16	Politician	B5	No Occupation

Table 4. Methods of expense for medical treatment

<i>ID_Pay</i>	<i>PAY_PTYPE_NAME</i>
01	General Patient
02	Patient with Social Security Card
03	Priest
05	Patient with Original Affiliation
06	Mahidol University Personnel
07	Faculty of Veterinary Med Personnel
08	Relatives
09	National Bank of Thailand Personnel
10	Special Patient
13	Golden Card from other Hospital
14	Golden Card from Sirirach Hospital
16	Government Official Rights of Continuous Treatment
S1	Patient with Social Security Rights from Sirirach Hospital
S2	Patient with Social Security Rights from Other Hospitals

4 Results

4.1 Data Mining Result

After analyzing diabetes mellitus (DM) patient data with Weka tool and applying association rules and Apriori Algorithm, the results are as following:

- Lower Bound M in Support is 0.1
- Min Metric = 0.9 quals
- NumRules = 20 rules

The results of data analysis are in Table5.

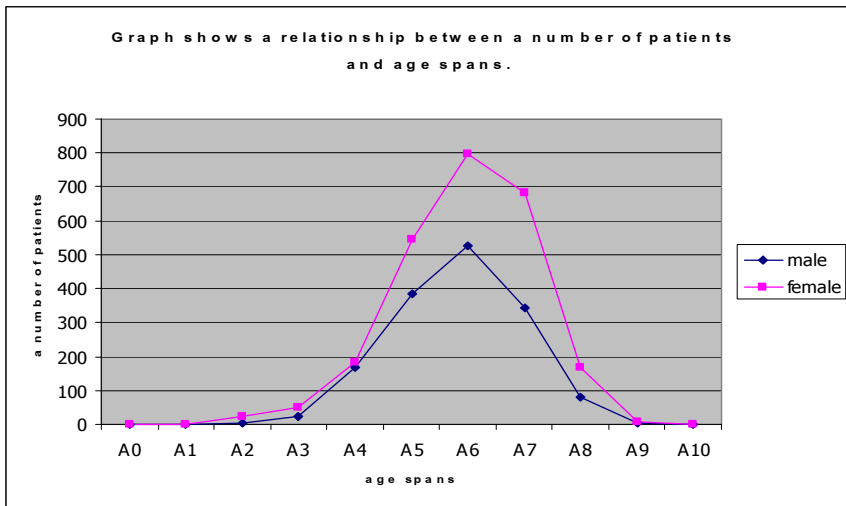
Interpretation of these 20 derived data concluded that both male and female diabetes mellitus(DM) patients aging between 60-69 years old who paid their treatment expense by Government Official Rights of Continuous Treatment tended to have diabetes mellitus(DM) with ophthalmic complication. The result is shown below Fig2, Fig3, and Fig4.

Table 5. Explanation of results derived from Apriori Algorithm Analysis

Best rules found:	Result Interpretation
1. A7=y 1016 ==> E113=y 1016 conf:(1)	DM patients type II aging between 70-79 years old with ophthalmic complications
2. A5=y 932 ==> E113=y 932 conf:(1)	DM patients type II aging between 50-59 years old with ophthalmic complications
3. female=y A7=y 677 ==> E113=y 677 conf:(1)	DM patients type II female aging between 70-79 years old with ophthalmic complications
4. male=y P16=y 664 ==> E113=y 664 conf:(1)	DM patients type II male Government Official Rights of Continuous Treatment with ophthalmic complications
5. O8=y 601 ==> E113=y 601 conf:(1)	DM patients type II Housework with ophthalmic complications
6. female=y A5=y 545 ==> E113=y 545 conf:(1)	DM patients type II female aging between 50-59 years old with ophthalmic complications
7. P16=y A7=y 538 ==> E113=y 538 conf:(1)	DM patients type II Government Official Rights of Continuous Treatment and aging between 70-79 years old with ophthalmic complications.
8. female=y O8=y 536 ==> E113=y 536 conf:(1)	DM patients type II female Housework with ophthalmic complications.
9. male=y A6=y 529 ==> E113=y 529 conf:(1)	DM patients type II male aging between 60-69 years old with ophthalmic complications .
10. male=y P1=y 410 ==> E113=y 410 conf:(1)	DM patients type II male General Patient with ophthalmic complications .
11. P1=y A6=y 403 ==> E113=y 403 conf:(1)	DM patients type II General Patient and aging between 60-69 years old with ophthalmic complications .
12. male=y 1534 ==> E113=y 1532 conf:(1)	DM patients type II female with ophthalmic complications
13. A6=y 1328 ==> E113=y 1326 conf:(1)	DM patients type II aging between 60-69 years old with ophthalmic complications
14. P16=y A6=y 559 ==> E113=y 558 conf:(1)	DM patients type II Government Official Rights of Continuous Treatment and aging between 60-69 years old with ophthalmic complications.

Table 5. (Continued)

15. P16=y 1628 ==> E113=y 1625 conf:(1)	DM patients type II Government Official Rights of Continuous Treatment with ophthalmic complications.
16. O5=y 518 ==> E113=y 517 conf:(1)	DM patients type II Trade with ophthalmic complications.
17. female=y A6=y 799 ==> E113=y 797 conf:(1)	DM patients type II female aging between 60-69 years old with ophthalmic complications.
18. P1=y 1140 ==> E113=y 1137 conf:(1)	DM patients type II General Patient with ophthalmic complications .
19. OB5=y 734 ==> E113=y 732 conf:(1)	DM patients type II No Occupation with ophthalmic complications .
20. female=y P16=y 964 ==> E113=y 961 conf:(1)	DM patients type II female Government Official Rights of Continuous Treatment with ophthalmic complications.

**Fig. 2.** Graph shows a relationship between diabetes mellitus (DM) patients with ophthalmic complication data and age spans

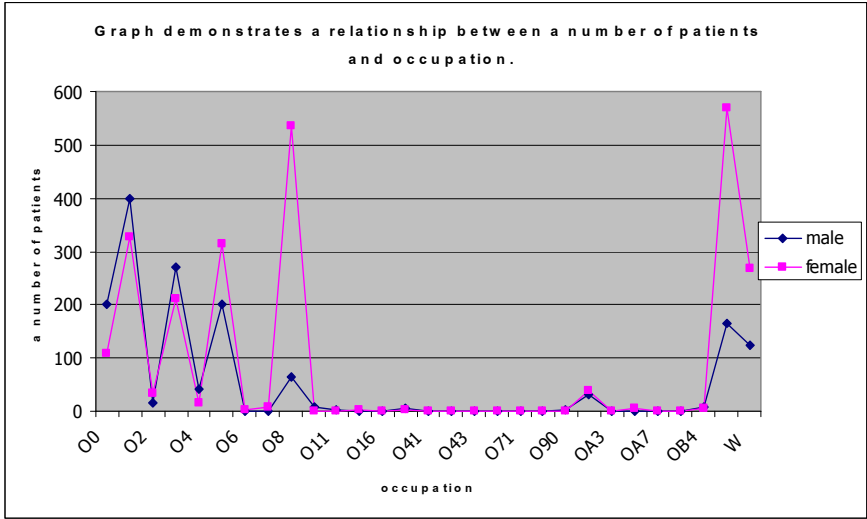


Fig. 3. Graph demonstrates a relationship between diabetes mellitus (DM) patients with ophthalmic complication data and occupation

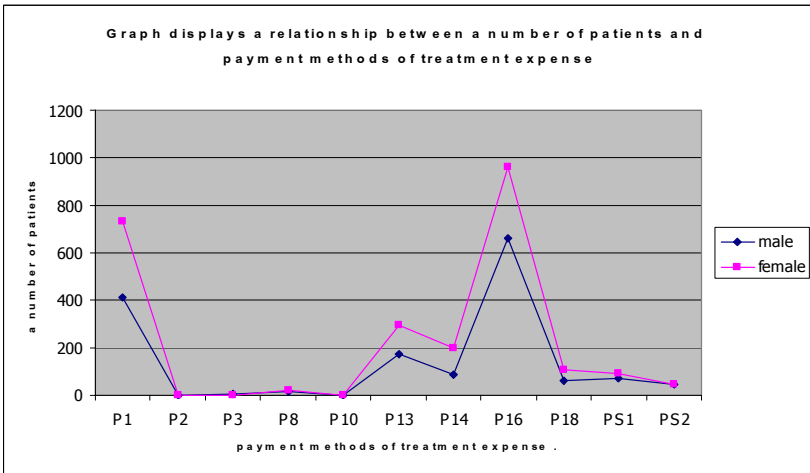


Fig. 4. Graph displays a relationship between diabetes mellitus (DM) patients with ophthalmic complication data and payment methods of treatment expense

5 Conclusion and Future Research

This research was a study of algorithm in searching for association rules of diabetes mellitus (DM) patient data with complication. The result of this study indicated that “both male and female diabetes mellitus (DM) patients Type II (Insulin-independent diabetes mellitus) aging between 60-69 years old with no occupation who paid their treatment expense by Government Official Rights of Continuous Treatment tended to

have diabetes mellitus (DM) with ophthalmic complication.” Ti proved that 60-69 years old population was in a retired – no occupation group. However, their payment method of treatment expense by Government Official Rights of Continuous Treatment verified their previous occupation as a government official who suffered from diabetes mellitus (DM) with ophthalmic complication.

In Thailand, data mining process still gained more consideration and became a useful tool in information management in healthcare environments. Immense service and clinical databases could be conveniently stored and recovered when needed. A part of computer program to produce distinct and useful information from databases was a key of data mining and the way to derive such a useful information was called learning algorithm. Computer programmers tested and compared the most proficient algorithm, but they still could not decisively indicate the most suitable one. It was because each set of information possessed distinctiveness, hence there was no best algorithm for all kinds of information.

6 Implication and Future Work

Data mining development and implication with medical can be accomplished in a range of proper techniques such as decision trees, classifier, clustering and association. Most suitable and efficient way of data mining will be deployed to predict medical data for the best practical treatment and useful for predicting other complication syndromes.

Data mining technique is a useful tool to analyze medical data in different aspects.

- Examine disease by analyzing patient symptoms such as initial lung and oral cancer which is difficult to discover by means of normal medical check. Genetic technology is helpful for faster and more accurate disease analysis.
- Predict disease progression: for example implication of biomarker helps a doctor to predict how long transplanted organ can be accepted by patient body.
- Proper treatment: It helps to anticipate treatment results.
- Understand disease mechanism in order to find out causes of disease for example a research of signaling pathway under virus infection state.

References

1. Chen, H., et al.: Medical Informatics: Knowledge discovery and data mining in medical informatics. Springer, New York (2005)
2. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press (1996)
3. Ng, R.T., Pei, J.: Special Issue: Data Mining for Health Informatics. ACM SIGKDD Exploration
4. Chao-ton, S., Chien-hsin, Y., Kuang-hung, H., Wen-ko, C.: Data Mining for the diagnosis of type II diabetes from three-dimensional body surface anthropometrical scanning data. Computer & Mathematics with Applications 51, 1075–1092 (2006)

5. Han, J., Rodriguze, J.C., Beheshti, M.: Diabetes Data Analysis and Prediction Model Discovery Using RapidMiner. In: 2008 Second International Conference on Future Generation Communication and Networking, pp. 69–99 (2008)
6. Zorman, M., Masud, G., Kokol, P., Yamamoto, R., Stiglic, B.: Mining Diabetes Database With Decision Trees and Association Rules. In: Proceedings of the 15th IEEE Symposium on Computer-Based Medical Systems (CBMS 2002), pp. 134–139 (2002)
7. Patil, B.M., Joshi, R.C., Toshniwal, D.: Association rule for classification of type-2 diabetic patients. In: 2010 Second International Conference on Machine Learning and Computing, pp. 330–334 (2010)
8. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: , Data Mining to Knowledge Discovery in Databases. American Association for Artificial Intelligence (1996)
9. Vision Problems in U.S.A. Statistical Analysis. National Society to Prevent Blindness, New York (1980)
10. Diabetic Retinopathy Study Research Group. Design, methods and baseline results. DRS report Number 6. Invest Ophthalmol. 21,149–209 (1981)

Intelligent Management Message Routing in Ubiquitous Sensor Networks

Junghoon Lee¹, Gyung-Leen Park¹, Hye-Jin Kim¹, Cheol Min Kim²,
Ho-Young Kwak³, Sang Joon Lee³, and Seongjun Lee⁴

¹ Dept. of Computer Science and Statistics,

² Dept. of Computer Education,

³ Dept. of Computer Engineering,

Jeju National University, 690-756, Jeju-Do, Republic of Korea

⁴ EZ Information Technology, Jeju Do, Republic of Korea

Abstract. This paper first presents an intelligent ubiquitous sensor network implementation for agricultural and livestock farms and designs an efficient cyclic routing scheme for network management messages, aiming at improving productivity and profit in those industries. Instead of multiple point-to-point message transmissions for status indication collection, our management message traverses the specific set of nodes of interest one by one. The management application collects routing information from the neighbor table commonly available in the current sensor protocol to calculate the communication cost between each pair of target nodes. Based on this topology view, a genetic algorithm solver decides the traversal sequence of a cyclic management path. The experiment result discovers that the multithreaded version, in which each thread runs its own initial population, can find a much better solution, efficiently escaping local traps.

Keywords: Ubiquitous sensor network, intelligent routing, management message, genetic algorithm, cost reduction.

1 Introduction

In the ubiquitous sensor network, or USN in short, a large number of nodes, having both computing power and wireless communication capability, are embedded in the environment to collect sensor data and report to the server [1]. USN is now being actively deployed to a wide range of application areas including monitoring space, monitoring things, and monitoring the interventions of things with each other [2]. Particularly, remote health control, military, inter-vehicle communication, building automation, and herd control are examples of most promising applications areas of USN [3]. Apparently, agricultural and livestock farms can also improve productivity and profit taking based on intelligent and efficient management, which is commonly achievable from mature sensor network technology [4]. In our USN implementation, the central controller monitors the status of crops and livestock using composite sensors developed by our own

design. A single sensor node can contain bio and environmental sensors, handling both data simultaneously.

A sensor network can be viewed as a large database system which responds to the query issued from various applications [5]. Practically, most modern sensor nodes work on TinyOS, which is a free and open source component-based operating system [6]. In addition, TinyDB extracts information from a network of TinyOS sensors just like the query processing system of the conventional database. It provides a simple, SQL-like interface to specify the data you want to extract, along with sensor-specific parameters such as data refresh rate [7]. In USN, queries on the sensor stream must run continuously over a given time amount to say nothing of the snapshot style information retrieval [8]. Hence, efficient management is essential in USN for monitoring, detecting, and dealing with a node or link failure.

The main task of USN monitoring is to collect information on node states such as battery level and communication power, network topology, wireless bandwidth, link state, and the coverage bound of USNs [9]. Monitoring individual nodes in a large sensor network may be impractical. It is sufficient to monitor and control the network just for the specific network coverage, namely, the availability of the specific set of nodes of interest. The manager node, or the coordinator, usually collects above-mentioned status information via point-to-point communication. However, the coordinator may need to know the availability of a path and whether a target set of nodes are all alive. In this case, instead of a series of point-to-point connections, a cyclic message relay that covers the target nodes looks advantageous in terms of message traffic and response time. Specifically, current sensor nodes commonly have neighbor table by which the message path can be decided either locally or globally.

In this regard, this paper is to present an intelligent USN architecture for agricultural and livestock farms first and then design an efficient routing scheme for management messages. To obviate overhead stemmed from multiple point-to-point transmissions [10], the routing message traverses the specific set of nodes and returns to the issuer. Upon the successful arrival of the message, the manager confirms the availability of path and that all nodes along the path are alive. Here, based on the global or subglobal view of network topology retrieved from the neighbor tables of involved nodes, the routing path can be decided by means of a genetic algorithm [11], which is an efficient search technique based on principles of natural selection and genetics.

The rest of this paper is organized as follows: After issuing the problem in Section 1, Section 2 describes the background and related work. Section 3 designs the cyclic routing scheme for management messages, and Section 4 measures the performance of the proposed scheme by means of prototype implementation. Finally, Section 5 concludes this paper with a brief introduction of future work.

2 Background and Related Work

Under a research and technical project named *Development of convergence techniques for agriculture, fisheries, and livestock industries based on the ubiquitous*

sensor networks, our project team has designed and built the prototype of an intelligent USN framework [4]. This framework provides an efficient and seamless runtime environment for a variety of monitoring-and-control applications on sensor networks. The sensor node, built on the Berkeley mote platform, comprises sensors, microprocessor, radio transceiver, and battery. Over the sensor network mainly exploiting the Zigbee technology, composite sensors detect events such as body heat change of a livestock via the biosensors as well as humidity, CO₂, and NH₃ level via the environmental sensors. Each node runs the IP-USN protocol and implements corresponding routing schemes. The sensor network and the global network, namely, the Internet, are connected through the USN gateway. At this stage, the system is to integrate a remote control model to provide remote irrigation and the activation of heater or fan. It must be mentioned that the feedback from farmers and farm owners keeps improving our system.

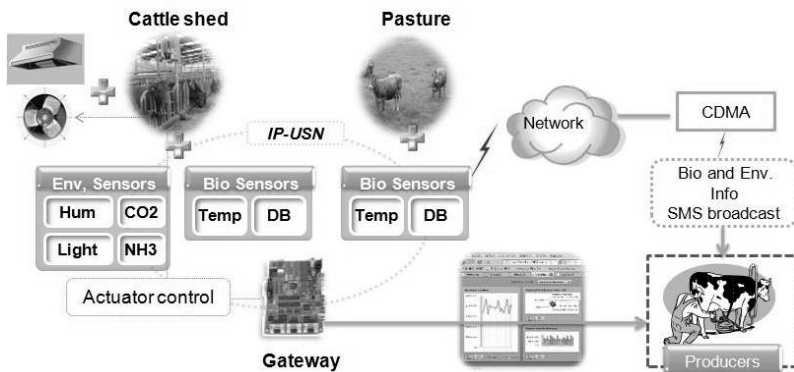


Fig. 1. Agricultural USN framework

Each sensor node installs TinyOS, which is an open source operating system designed for low-power wireless devices, ubiquitous computing, personal area networks, and so on. Particularly, TinyOS has been adopted by a broad range of wireless sensor networks. Its communication-centric design and modular software model are tailored to the unique requirements of respective networks, where applications and services are distributed over collections of resource-constrained, unattended application-specific devices streaming data to and from the physical world. TinyDB is a distributed query processor that runs on each sensor nodes over the whole network. TinyDB runs on top of the TinyOS operating system, basically providing traditional primitives such as *select*, *join*, *project*, and *aggregate data* [12]. The main component of TinyDB necessarily includes a small query processor, which fetches the values of local attributes, receives sensor readings from neighboring nodes, combines and aggregates these values together, filters out undesired data, and outputs values to parents. TinyDB manages the underlying radio network by tracking neighbors as well as maintaining routing tables.

Our previous work has designed an intelligent data processing framework in ubiquitous sensor networks, implementing its prototype [4]. Much focus is put on how to handle the sensor data stream as well as the interoperability between the low-level sensor data and application clients. This work first designs systematic middleware which mitigates the interaction between the application layer and low-level sensors, for the sake of analyzing a great volume of sensor data by filtering and integrating to create value-added context information. Then, an agent-based architecture is proposed for real-time data distribution to forward a specific event to the appropriate application, which is registered in the directory service via the open interface. The prototype implementation demonstrates that this framework can not only host a sophisticated application on USN and but also autonomously evolve to new middleware, taking advantages of promising technologies such as software agents, XML, and the like. Particularly, cloud computing can provide the high-speed data processing framework for sensor streams [13].

3 Management Message Delivery Scheme

Each sensor node installs SP (Sensornet Protocol), which works between the network layer and the link layer, providing a link abstraction as shown in Figure 2 [3]. The data transfer of SP is quite similar to acknowledged connectionless communication while the network layer can specify per-packet priority and reliability flag. Priority field decides the transmission order in the message pool. If the reliability flag is set in a message, SP makes the link protocol initiate an available reliability mechanism in packet transmission. For example, the link layer may request acknowledgement from the recipient and retry an unacknowledged packet. In addition to single packet sending, SP also supports *message futures* to set the number of packets awaiting to be sent. According to this value, SP may fetch the remaining unsent packets quickly. In the mean time, the network layer automatically receives feedback to each packet sending request. Namely, the network layer application gets notified of whether the packet was transmitted successfully and whether the underlying channel is congested.

Maintaining the state of each neighbor is necessary to achieve reliability and efficiency in data transfer. In SP, each local node keeps track of its neighbors' schedules in order to know when the link to a specific neighbor will become available. To this end, each record in its neighbor table entry includes neighbor ID, time-on (local time when neighbor will wake up), time-off (local time when neighbor will go to sleep), listen (listen to neighbor during its next wakeup period), quality (link quality metric), and the like. SP constantly updates neighbor schedules using the time-on and time-off fields in a neighbor table entry. The link protocol refers to this information to schedule packet transmission. A network layer protocol can set a neighbor's listen flag to make the local node wake up and listen to the specific neighbor on its next wakeup period.

The SP neighbor interface basically defines several primitive interfaces to add, remove, and update an entry. Using the neighbor table and the exchange of

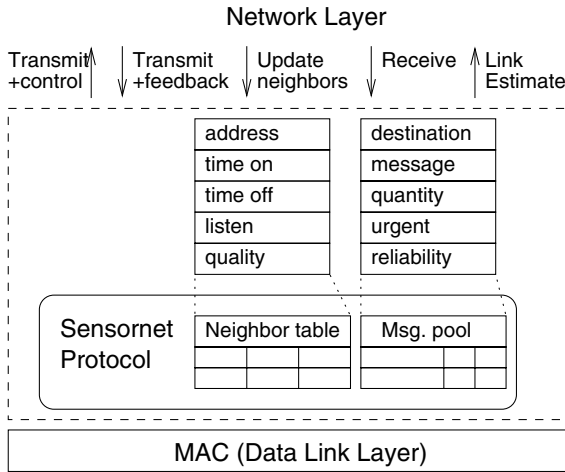


Fig. 2. Sensornet layers

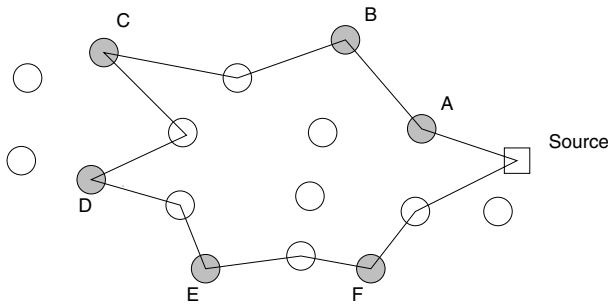


Fig. 3. Management message routing

routing information, each node can decide the cost, namely, the delay to the other nodes in the target management group. The manager node, marked as source in Figure 3, can know the delay between every pair of nodes by periodic message exchange. A management message is triggered from the manager node and the message contains the order it will visit. Receiving this message, a node forwards it to the next node. During the traversal, the message may experience retransmission. On the return of the message, the manager knows the path and node condition based on the traversal time or possibly from the predefined status indication in the message payload.

The sequence decision is quite similar to the traveling salesman problem for which a lot of optimization algorithms have been designed to get optimal and suboptimal solutions. Among these, the genetic algorithm can find the suboptimal path in a tunable execution time. Genetic algorithms are efficient search techniques based on principles of natural selection and genetics. They have been

successfully applied to find acceptable solutions to problems in business, engineering, and science within a reasonable amount of time [11]. Each evolutionary step generates a population of candidate solutions and evaluates the population according to a fitness function to select the best solution and mate to form the next generation. At each evolution, bad traits are eliminated while the good traits survive and are combined with other good traits to make better candidates. Over a number of generations, good traits dominate the population, resulting in an increase in the quality of the solutions.

The route between the target nodes is represented by an array where each element corresponds to a member node. Each chromosome contains every node to visit exactly once. The first step of the genetic algorithm is the random generation of the initial population, and then it iterates evaluation of objective functions, reproduction of population, crossover, and mutation. A simple crossover reproduction is not valid as it makes the illegal chromosome, namely, some nodes may appear more than once. To overcome this problem, we implement a modified reproduction. After generating a new chromosome from two parents, the controller invalidates duplicated elements from the route, identifying the missed nodes at the same time. The router replaces each invalidated element with a node randomly chosen from the missed node list. This step makes the chromosome consistent to the routing schedule.

In addition, mutation exchanges elements randomly within a chromosome to prevent the search procedure to be trapped in a local minimum. Generally, the search procedure sets the mandatory rate of mutation. However, our implementation observes that many identical genes are generated in a population. Hence, every time a gene is generated, the procedure checks if it is already included in the population. If so, the procedure attempts to conduct mutation until finding a new chromosome completely different from the existing ones.

4 Performance Measurement

This section implements the proposed management routing scheme in USN with Visual C++ 6.0, making it run on the platform equipped with Intel Core2 Duo CPU, 3.0 GB memory, and Windows Vista operating system. In our implementation, the number of genes in each population is set to 25. For the random generation of an initial population, the system clock value is used for the seed of random number setup. The first experiment measures the minimum cost change according to the progress of iteration steps. Figure 4 plots 3 curves, each of which has its own initial generation. As shown in the figure, each stream begins from the high cost solutions, but converges to a constant value soon. 3 Streams find their own message routes having costs of 43, 47, and 39, respectively.

Once the cost reaches the constant value, the subsequent iterations can hardly achieve further reduction. Hence, the next experiment measures the average cost of all routes in the population along with the progress of iterations. It is clear that the quality of new-generation genes greatly depends on that of parent genes, which are selected for mating in generating new genes. As shown in Figure 5, the

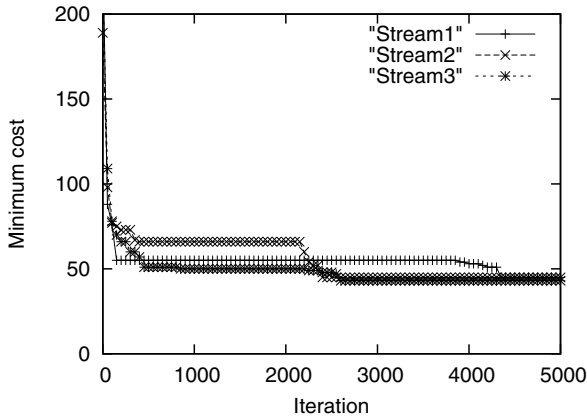


Fig. 4. Cost reduction

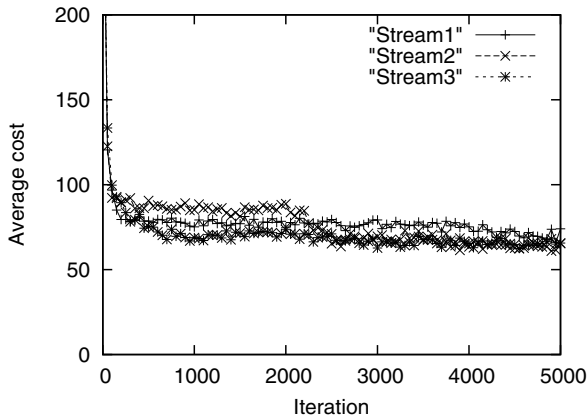


Fig. 5. Average cost for each population

average cost reduces slowly as the iteration proceeds. In spite of the improvement in the average cost, the minimum cost gets hardly improved.

Previous experiments indicate the importance of initial set generation. Hence, we measure the performance of the threaded generation scheme, where each thread runs with its own initial population. Nowadays, threaded execution is very popular even in personal computers and hand-held devices. It can benefit from parallel processing on the multiprocessor architecture, significantly enhancing the computing time. Figure 6 plots the result and the controller can find the route having cost of 39 with 10 threads.

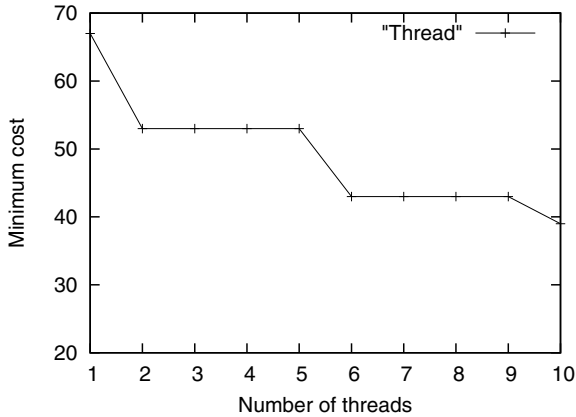


Fig. 6. Effect of multi-threaded search

5 Concluding Remarks

This paper has first presented an intelligent ubiquitous sensor network implementation for agricultural and livestock farms and designed an efficient cyclic routing scheme for network management messages to improve the overhead and response time of a sensor network management application. Instead of multiple point-to-point exchanges of management messages, our scheme issues a message which will traverse the specific set of nodes of interest one by one. The path decision for the management message depends on the neighbor node table stored in each sensor node. Based on communication cost between each pair of target nodes, a genetic algorithm solver is implemented to decide the traversal order of a cyclic management path with a tunable number of iterations, namely, within a specific execution time bound. The experiment result discovers that the multithreaded version, in which each thread runs its own initial population, can always find a much better solution, efficiently escaping local traps.

As future work, we are planning to design a data mining tool for management information as well as sensor data [14]. The sophisticated data analysis will create a new type of management messages and those messages will make USN more intelligent.

Acknowledgments. This research was supported by the MKE (The Ministry of Knowledge Economy), through the project of Region technical renovation, Republic of Korea.

References

1. Lee, J., Park, G., Rehman, S., Jhang, S., Kang, M.: A Real-time Message Scheduler Support for Dual-Sink Mobile Ad-hoc Sensor Networks. In: 24th Annual ACM Symposium on Applied Computing, pp. 305–309 (2009)

2. Culler, D., Estrin, D., Srivastava, M.: Overview of Sensor Networks. *IEEE Computer* 37, 41–49 (2004)
3. Cuevas, A., Urueña, M., Laube, A., Gomez, L.: LWESP: Light-Weight Exterior Sensornet Protocol. In: *IEEE International Conference in Computer and Communications* (2009)
4. Lee, J., Park, G., Kwak, H., Kim, C.: Efficient and Extensible Data Processing Framework in Ubiquitous Sensor Networks. In: *International Conference on Intelligent Control Systems Engineering*, pp. 324–327 (2011)
5. Madden, S., Franklin, M.: Fjording the Stream: An Architecture for Queries over Streaming Sensor Data. In: *Proc. of the 2002 Intl. Conf. on Data Engineering* (2002)
6. <http://www.tinyos.net>
7. Golab, L., Oszu, M.: Issues in Data Stream Management. *ACM SIGMOD Record* 32, 5–14 (2003)
8. Madden, S., Franklin, M., Hellerstein, J., Hong, W.: The Design of an Acquisitional Query Processor for Sensor Networks. *ACM SINGMOD* (2003)
9. Ruiz, L., Nogueira, J., Loureiro, A.: MANNA: a Management Architecture for Wireless Sensor Networks. *IEEE Communications Magazine* 41, 116–125 (2003)
10. Lee, J., Song, H., Mok, A.: Design of a Reliable Communication System for Grid-Style Traffic Control Networks. In: *The 16th IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 133–142 (2010)
11. Katsigiannis, Y., Georgilakis, P., Karapidakis, E.: Multiobjective Genetic Algorithm Solution to the Optimum Economic and Environmental Performance Problem of Small Autonomous Hybrid Power Systems with Renewables. In: *IET Renewable Power Generation*, pp. 404–419 (2010)
12. Madden, S., Franklin, M., Hellerstein, J., Hong, W.: TinyDB: an Acquisitional Query Processing System for Sensor Networks. *ACM Transactions on Database Systems* 30 (2005)
13. Kang, M., Kang, D., Crago, S., Park, G., Lee, J.: Design and Development of a Run-time Monitor for Multi-Core Architectures in Cloud Computing. *Sensors* 11, 3595–3610 (2011)
14. Woo, H., Mok, A.: Real-Time Monitoring of Uncertain Data Streams Using Probabilistic Similarity. In: *Proc. of IEEE Real-Time Systems Symposium*, pp. 288–300 (2007)

On Ranking Production Rules for Rule-Based Systems with Uncertainty

Beata Jankowska¹ and Magdalena Szymkowiak²

¹ Institute of Control and Information Engineering

² Institute of Mathematics,

Poznan University of Technology, Pl. M.Sklodowskiej-Curie 5, 60-965 Poznan, Poland

{beata.jankowska,magdalena.szymkowiak}@put.poznan.pl

Abstract. There are many places (e.g. hospital emergency rooms) where reliable diagnostic systems might support people in their work. The paper discusses the problem of designing diagnostic rule-based systems with uncertainty. Most such systems use the technique of forward chaining in their reasonings. The number and the contents of the hypotheses depend then on both the form of system's knowledge base and the details of the inference engine performance. In particular, the hypotheses can be influenced by the rules' priorities. In the paper we propose a method for determining priorities for the rules designed from true evidence base which contains aggregate data of an attributive representation.

Keywords: rule-based system, uncertainty, attributive data.

1 Introduction

In the last years there has been an increasing interest in designing rule-based systems (RBSs) [1] as systems strictly cooperating with reasoners for logics (mainly – Description Logics, eg. [2] or Attributive Logics, eg. [3]). After having implanted appropriate ontological knowledge, an RBS increases its expressive power and usefulness for supporting the processes of diagnostics and classification in the domain.

In order to be used in empirical domains, expert systems should implement the notion of ‘uncertainty’. As concerns RBSs, they do have to be RBSs with uncertainty, that enable representing uncertain knowledge and making uncertain reasonings. Also these systems can be designed using domain ontological knowledge, which improves the quality of their knowledge bases and the effectiveness of reasonings [4].

However, the RBSs with uncertainty need using special means that can express the level of certainty, with reference to both data (facts) and knowledge (rules). The most often used means are fuzzy notions [5] (e.g. `high_temperature`, `elderly_patient`), ignorance ranges [6] (e.g. `brain_tumor <0.1, 0.3>`), or certainty factors [7] (`brain_tumor CF 0.15`).

The main methods of reasoning in RBSs are forward chaining and backward chaining. The first one is being done with a view to reproducing knowledge, while the latter one – with a view to proving a given hypothesis. A typical RBS use consists in applying forward chaining, optionally – under limitation of the solution space, e.g. by means of Magic Transformation [8]. Focusing our attention on that type of inference engine, let us remind that, in RBSs not being locally confluent, the final results of reasonings depend,

in general, on the order of active rules firing. This order depends, in turn, on the algorithm of agenda conflict resolution which often prefers rules of highest priorities. For this reason, the process of establishing rules' priorities should be carried out very carefully.

Let us observe that most diagnostic procedures and systems have tendency to prefer those dependencies which prove strong relations between antecedents and consequents. Such a requirement is fundamental, among others, for designing association rules [9]. These rules are being created under the constraint that confidence (i.e. the level of certainty of rule's conclusions given the occurrence of all its premises) is not less than a required threshold. Meanwhile, the general underestimating of weak dependencies is not a good approach. The following medical examples prove the truthfulness of this remark.

It is common medical practice that before administering a pharmacological treatment, a doctor asks about all possible effects of this treatment in patients, both proper ones and adverse ones. The latter effects might make the treatment difficult or even impossible. If taking an oral antibiotic X, used with success in the treatment of sore diseases, causes an adverse effect of torsions in 5% cases only, then one can consider it as an appropriate drug for a physically weakened man with purulent tonsillitis. If this percentage was much higher than reported, then the doctor should think of administering some other drug to the patient, causing less adverse effects compared with the considered antibiotic X. To conclude, the hypothesis about sporadic occurring of torsions as an effect of taking X is important **just for the sake of a weak dependence** of torsions on taking X.

Let us now consider a medical case similar to the one quoted by Zadeh in his classical paper on fuzzy sets. Assume that in a sick room a boy, aged 14 years, suffering from strong vertiges, nausea and spatial orientation disturbances, is being seen by a surgeon on duty. If the doctor knows nothing more about him, then the boy will be diagnosed – with probability about 50% – with a state after 'brain concussion', and – with a similar probability – with 'meningitis'. However, reliable medical evidence confirms the fact that in such a case, in 1% of teenage boys, the correct diagnosis is 'brain tumor'. In such a situation, it is purposeful to consider this particular hypothesis as a first one, in order to confirm/exclude it and make an immediate decision about surgical intervention. To conclude, the considered hypothesis is worth checking **even though there is only a weak dependence** of 'brain tumor' on the observed symptoms.

The above examples confirm us in the belief that the rules' priorities should depend on more than strong relationships between their premises and conclusions. These priorities are influenced by a number of important factors. In [4] an algorithm for designing rules with uncertainty has been proposed. It is based on machine learning from attributive data with set values. Using the knowledge of domain taxonomy and applying semantic data integration, the algorithm produces a great number of reliable rules with uncertainty. The quoted paper does not present the method of calculating rules' priorities. It was the subject of our last research and it is the topic of the present paper.

The contents of the paper is as follows. Chapter 2 describes a new model of RBS with uncertainty, partially presented in [4]. The rules are defined using so called reliability factors, taking values from the range $<0;1>$ and representing empirical levels of certainty. In this chapter also the problem of propagating uncertainty is shortly discussed. The method of determining the certainty of the rule's conclusion given certain occurrences of the rule's premises is presented in Chapter 3, while the method of determining the certainty of the rule, taken as a whole, given stored evidence – in Chapter 4. Chapter 5 contains final conclusions. All the considerations are illustrated by means of medical examples.

2 Rule-Based Systems with Uncertainty

From the point of view of the theory of probability, an RBS with uncertainty can be seen as a set of rules with attached levels of conditional probabilities (a fact with uncertainty can be considered as a kind of rule with a premise representing user's knowledge and the fact in the role of conclusion). In order to determine the mentioned levels, one can use ignorance ranges [6], being defined by means of two conditional probabilities: a minimal one, representing so called necessity level (Bel), and a maximal one, representing so called plausibility level (Pls). The range $\langle Bel; Pls \rangle$ should contain an actual value of the probability.

In turn, certainty factors, proposed by the authors of MYCIN [7], inform us not about exactly the level of conditional probability, but rather about the level of increase of the probability of the rule's conclusion given occurrences of the rule's premises. They inform about it by means of one number coming from the range $\langle -1; 1 \rangle$, where 1 and -1 stand for that the conclusion is certainly true and certainly false, respectively, given occurrences of the premises. The functions for propagating uncertainty through reasoning chains [10] enable to update the values of certainty factors while performing reasoning.

Appreciating certainty factor as precise and general method of stating the level of conditional probability we, however, see the incompatibility of its measure with the measure of classical probability (the range $\langle 0; 1 \rangle$). Besides, we have reservations about the combination function for multiple rules concluding the same conclusion. It is correct only when premises from all such rules are independent. The constraint cannot be expected to hold in general, in particular – in an RBS designed from true evidence base.

In the light of these comments, we propose yet another method to represent the level of conditional probability. We mean here so called 'reliability factor', a sole number coming (implicitly) from the range $\langle Bel; Pls \rangle$, and representing the empirical probability obtained by real data analysis.

2.1 Rule with Uncertainty – Syntax and Semantics

Let \mathbf{S} stand for a conceptualized domain and $\mathbf{O}_{\mathbf{S}}$ represents its ontology. Besides, let $\mathbf{A}_{\mathbf{S}}$ stand for a set of concepts from $\mathbf{O}_{\mathbf{S}}$, chosen arbitrarily in such a way to reflect the state of some object (e.g. the results of a medical study). Under these assumptions, by a rule with uncertainty we will mean an implication of the following form:

$$\begin{aligned} &\text{it happens with } grf = p_r : \\ &\text{if } P_1 \text{ and } P_2 \text{ and } \dots \text{ and } P_n \\ &\text{then } C \text{ with } irf = p_c, \end{aligned} \quad (1)$$

where P_i , $1 \leq i \leq n$, and C stand for facts of the form $A = V_A q_A$, with $A \in \mathbf{A}_{\mathbf{S}}$, called as 'attribute A ', V_A being a set of subconcepts of A ; $q_A \in \{\odot, \oplus\}$, giving an interpretation for set V_A that should be understood as: \odot – conjunction of all subconcepts from V_A , \oplus – disjunction of all subconcepts from V_A ; grf stands for the name of *global reliability factor*; irf – for the name of *internal reliability factor*; $p_r, p_c \in \langle 0; 1 \rangle$ – for the values of factors grf and irf , respectively.

In the rule with uncertainty (1), fact-conclusion C is conditionally dependent on fact-premises P_1, P_2, \dots, P_n . Value p_c of the rule's irf represents the probability of occurrence of conclusion C given certain occurrences of all premises P_1, P_2, \dots, P_n .

Besides, the rule with uncertainty (1), taken as a whole, is conditionally dependent on the contents of an evidence base from which it was derived. Value p_r of the rule's grf represents the probability of that the rule is fully reliable given the evidence. While it is true that factor grf does not directly influence the certainty of hypothesis C, however it can really influence the course and the final result of reasoning.

The semantics of factors irf and grf and the methods of their calculation are widely discussed in Chapters 3 and 4, respectively.

Here is an exemplary rule with uncertainty r_1 , representing a hypothesis on 'brain tumor' made for teenage boys diagnosed in the way reported in Chapter 1.

```
r1 : it happens with grf = 0.82 :
    if Current_symptoms = {vertiges, nausea, visual_disturbances}⊙ and
       Gender = {male}⊙ and
       Age_range = {11,...,18} ⊕
    then diagnosis = {brain_tumor}⊙ with irf = 0.01
```

Co-occurrence of the low irf value 0.01 and the high grf value 0.82 deserve underlining. The factors prove that hypothesis generated by the rule – although specific – is very reliable.

2.2 Uncertain Reasoning

As it was said before, we consider such RBSs with uncertainty that perform forward chaining. Then, the final result of reasoning depends as well on the state of RBS's knowledge base (rules) and RBS's data base (facts), as on the algorithm of agenda conflict resolution and the functions for propagating uncertainty through reasoning chains. Here we assume that the choice of an active rule from agenda is being done based the rule's priority, which depends – in the first place – on the rule's grf (see Sec. 4.4).

As regards the functions for propagating uncertainty, all of them but the combination function for multiple production rules concluding the same conclusion will be similar to their equivalents from [10]. For example, let us here define the combination function for propagating uncertain evidence. Let facts P_1, P_2, \dots, P_n be present in RBS database and provided with irf values p_1, p_2, \dots, p_n , respectively, at a moment. Then, a rule of the form (1) will be fired only if the relation $\min\{p_1, p_2, \dots, p_n\} \cdot pR \geq \tau$ holds, where τ stands for a required threshold. If it happens, then the fact C concluded by the rule will be provided with the following new value of irf :

$$irf = \min\{p_1, p_2, \dots, p_n\} \cdot pC \quad (2)$$

instead of the previous value: $irf = pC$.

Let us now consider in detail the combination function for multiple rules concluding the same conclusion. First, let us remark that the assumption on the independence of the premises from those multiple rules does not have to be fulfilled, especially – in RBSs based on true evidence base. In general, it would be very difficult to check such an independency (the algorithm proposed in [4] takes care only of consistency/nonredundancy of RBS knowledge base being designed).

The combination function used in MYCIN takes all multiple rules as they were equally reliable. We will replace it by a function that differentiates rules with respect to their reliabilities. Its definition is based on the assumption that the rules of low grf have been judged as secondary compared with the others of high grf . Their presence in the agenda can be only a consequence of putting some facts into the system's database by any of preceding rules. That is why, successive irf values should be given weights, decreasing with the progress of reasoning. The demand is fulfilled by the following overloaded function f for calculating irf of hypothesis C concluded by multiple rules:

$$\begin{aligned}
 \text{irf} &= f(C, \mathbf{KB}) = f(C, \mathbf{KB}, d) && \text{where} \\
 f(C, \mathbf{KB}, i) &= \begin{cases} v_1 \cdot \text{irf}_{r1} & \text{for } i = 1 \\ (1 - v_i) \cdot f(C, \mathbf{KB}, i - 1) + v_i \cdot \text{irf}_{ri} & \text{for } 2 \leq i \leq d \end{cases}, && (3)
 \end{aligned}$$

where **KB** stands for a knowledge base containing *n* rules with uncertainty; $d \leq n$ – for a number of rules r_i (from among all rules fired in the current process of reasoning) that concluded hypothesis *C*; irf_{ri} ($1 \leq i \leq d$) – for *irf* value of conclusion *C* obtained after having fired rule r_i ; and v_i ($1 \leq i \leq d$) – for a dynamically calculated weight of irf_{ri} :

$$v_i = \begin{cases} 1 & \text{for } i = 1 \\ \frac{v_{i-1}}{t + v_{i-1}} & \text{for } 2 \leq i \leq d \end{cases} \quad (4)$$

where *t* is a constant of proportion between the weights of *irf* from two successive rules concluding hypothesis *H*. For further considerations, we propose to use $t = 1.1$.

To realize the importance of factor *grf*, let us analyze an example of diagnostic reasoning with reference to the boy described in Chapter 1. Differently than previously, let us assume that he complained of a headache. Let us suppose that reasoning is in progress at the moment and none of rules concluding *brain_tumor* has been fired up to this moment. If rules r_1 (from Chapter 1) and r_2 (the one given below) were now successively fired (based on the rules’ activities and the advantage of $\text{grf}_{r1}=0.82$ over $\text{grf}_{r2}=0.74$):

```

r2 : it happens with grf = 0.74 :
    if Current_symptoms = {vertiges, visual_disturbances}⊙ and
       Age_range = {10,...,100}⊕ and
       Anamnesis_result = {headache}⊙
    then diagnosis = {brain_tumor}⊙ with irf = 0.12
    
```

then, given certain occurrences of all premises from rules r_1 and r_2 , after firing these two rules, hypothesis *Diagnosis* would be as follows:

Diagnosis = *brain_tumor* with *irf* = 0.0624 .

However, if only we swapped these two factors *grf* and, consequently, reversed the order of rules firing, then hypothesis *Diagnosis* would be as follows:

Diagnosis = *brain_tumor* with *irf* = 0.0676 .

The obtained *irf* values differ by less than 10%. However, if we used a higher constant of proportion, e.g. $t = 1.5$ or $t = 2.0$, then the difference would be much higher (nearly 40% and 80%, respectively). Regardless of the value of constant *t*, factor *grf* can efficiently differentiate all the rules with respect to their reliability and importance (see Chapter 4).

3 Internal Rule’s Reliability

Our proposal of designing diagnostic rules with uncertainty [4] is based on the exploration of data contained in a true evidence base. We assume that these are aggregate data. In medicine, chosen as an exemplary domain, such data characterize not an individual patient but a group of patients. The first idea concerning the calculation of reliability factors of rules obtained from individual patients’ data was presented in [11]. A detailed description of methods for calculating these factors for rules obtained from aggregate data is the subject of our present considerations.

To shortly summarize the algorithm for designing rules with uncertainty, we will use a medical example. The following tuple T_1 represents aggregate data characterizing a group of 54 young patients hospitalized for bronchial asthma [12]:

$T_1 = \langle \text{General_Diagnosis}=\{\text{pediatric_asthma}\} \odot /54,$
 $\text{Drug}=\{\text{short-act_beta2_agonist, inhaled_anticholin}\} \odot /54,$
 $\text{co_intervention}=\{\text{systemic_corticosteroid}\} \odot /54,$
 $\text{age_range}=\{1, \dots, 7\} \oplus /54,$
 $\text{symptoms}=\{\text{coughing, wheezing}\} \odot /54,$
 $\text{treatment_effects}=\{\text{no_hospital_admis}\} \odot /29,$
 $\text{adverse_effects}=\{\text{vomiting}\} \odot /2 \rangle .$

Without going into details, let us only say that attributes written in capital letters were regarded as key ones while selecting patients to this group. As a result of integration of tuple T_1 with other similar tuples we could obtain the following virtual tuple T :

$T = \langle \text{General_Diagnosis}=\{\text{pediatric_asthma}\} \odot /231,$
 $\text{Drug}=\{\text{short-act_beta2_agonist, inhaled_anticholin}\} \odot /231,$
 $\text{age_range}=\{1, \dots, 18\} \oplus /231,$
 $\text{symptoms}=\{\text{coughing}\} \odot /231,$
 $\text{treatment_effects}=\{\text{no_hospital_admis}\} \odot /139$
 $\text{adverse_effects}=\{\text{vomiting}\} \odot /5 \rangle .$

The number of patients ‘caught’ in this tuple is equal 231. All the patients were the same with respect to General_Diagnosis, Drug, age_range and symptoms (common attributes), and they differed with respect to treatment_effects and adverse_effects (discriminatory attributes). ‘Attribute_counts’ of the discriminatory attributes treatment_effects = {no_hospital_admis} \odot and adverse_effects = {vomiting} \odot are equal 139 and 5, respectively. For virtual tuple T , the following rules r_3 and r_4 could be obtained:

r_3 : it happens with grf = 0.87:

if General_Diagnosis =
 {pediatric_asthma} \odot and
 Drug = {short-act_beta2_agonist,
 inhaled_anticholin} \odot and
 age_range = {1, ..., 18} \oplus and
 symptoms = {coughing} \odot
 then treatment_effects =
 {no_hospital_admis} \odot with irf = 0.6

r_4 : it happens with grf = 0.9 :

if General_Diagnosis =
 {pediatric_asthma} \odot and
 Drug = {short-act_beta2_agonist,
 inhaled_anticholin} \odot and
 age_range = {1, ..., 18} \oplus and
 symptoms = {coughing} \odot
 then adverse_effects =
 {vomiting} \odot with irf = 0.02

Further on, we will discuss the calculation of factor irf and factor grf for the above rules. As it was said before, these factors will decide about the priorities of rules in the designed RBS knowledge base.

Let us assume that a rule with uncertainty was obtained from a virtual tuple T in which ‘attribute_count’ of common attributes (corresponding to the premises of the rule) is equal N , and ‘attribute_count’ of the chosen discriminate attribute (corresponding to the conclusion of the rule) is equal L (where $L \leq N$). Then, *internal reliability factor* of the rule can be calculated from the formula:

$$\text{irf} = \frac{L}{N} \cdot \quad (5)$$

Factor irf takes its value from the range $\langle 0; 1 \rangle$ and it is the counterpart of the confidence from association rules [9]. In statistics, it is the counterpart of the point estimate of the proportion corresponding to the conditional probability of the rule’s

conclusion, given the certain occurrence of the rule's premises [13]. High (close to 1) level of *irf* will be typical of rules with a highly probable conclusion. Obviously, such rules should have high priorities in the knowledge base of RBS. However, we suggest that a low (close to 0) level of *irf* should also influence a high priority of the rule. It is so because a low probability of the fact stated in the rule's conclusion implies a high probability of the opposite one. A rule with extreme *irf* (close to 1 or close to 0) will be regarded as a rule with a 'characteristic' conclusion.

For exemplary rules *r3* and *r4*, 'attribute_count' *N* of common attributes is equal 231 and 'attribute_counts' *L₄* and *L₅* of the chosen discriminate attributes are equal 139 and 5, respectively. Then, their internal reliability factors, calculated from formula (5), are equal:

$$\text{irf}_{r3} = \frac{139}{231} = 0.6 \quad \text{and} \quad \text{irf}_{r4} = \frac{5}{231} = 0.02, \text{ respectively.}$$

It means rule *r3* has a higher internal reliability factor compared to rule *r4* but it is not equivalent to its having a higher priority in the designed knowledge base. Let us notice that the conclusion of rule *r3* is less 'characteristic' than the conclusion of rule *r4*. In Chapter 4, factor *grf* will be defined. It is this factor that really determines the rule's priority.

4 Global Rule's Reliability

The problem of determining *global reliability factor* *grf* seems to be very complex. To solve it correctly, it is necessary to take a few different parameters into account. In our opinion, among others these should be: a rule's *weight* *wg* and a rule's *accuracy* *ac*. We propose to estimate factor *grf* using the following formula:

$$\text{grf} = \min \{wg, ac\}. \quad (6)$$

It means that a rule will be considered to have a high level of global reliability factor, if it has, at the same time, a high weight and a high accuracy.

4.1 Rule's Weight

The parameter that should have an influence on factor *grf* is a rule's weight *wg*. The value of this parameter will depend mainly on number *N* standing for 'attribute_count' of common attributes in a virtual tuple *T*, being the base for rule designing.

To determine a rule's weight *wg*, we will firstly estimate a 100%·(1- α) confidence interval for factor *irf* being a proportion which corresponds to the conditional probability of the rule's conclusion given the certain occurrence of its premises. For a given large enough *N* ($N \geq 100$, [13]), we can assume that factor *irf* has an asymptotically standard normal distribution and the length of its confidence interval can be calculated from the following formula:

$$l_{1-\alpha} = 2 \cdot u_{1-\frac{\alpha}{2}} \cdot \sqrt{\frac{\text{irf} \cdot (1-\text{irf})}{N}} \quad (7)$$

We consider it proper to make a rule's weight *wg* dependent on the exactness of the interval estimation. This exactness is considered to be the result of subtraction $1 - l_{1-\alpha}$. Since the estimation refers to the proportion, then the confidence interval will be limited to interval $\langle 0; 1 \rangle$, and then, its length $l_{1-\alpha}$, and also exactness $1 - l_{1-\alpha}$, will take

their values from the range $<0; 1>$. In practice, we usually estimate confidence intervals of 95%. In such a case, the required critical value of the standard normal distribution is equal $u_{0.975} = 1.96$, and exactness $1 - l_{0.95}$ takes its value from the range $<0; 1>$ for each N greater or equal 4, $N \geq 4$. Then, taking into account the initial assumption that $N \geq 100$, we can define a rule's *weight* wg as follows:

$$wg = \min\{1 - l_{0.95}, 0.95\}. \tag{8}$$

The higher number N , the smaller length $l_{0.95}$; the smaller length $l_{0.95}$, the higher exactness $1 - l_{0.95}$; the higher exactness $1 - l_{0.95}$, the higher weight wg . Hence, we can state that the process of data integration can significantly increase a rule's weight wg .

However, let us notice (see formula (7)) that exactness $1 - l_{0.95}$ depends not only on number N but also on the extremity of factor irf : the more extreme (close to 1 or close to 0) factor irf , the smaller length $l_{0.95}$ and the higher exactness $1 - l_{0.95}$. It means that the rule's weight is higher if the conclusion of the rule is more 'characteristic'.

The last parameter having an influence on the rule's weight wg is the confidence level of estimation. We should remember that the confidence interval includes estimated factor irf with probability not exceeding $1 - \alpha = 0.95$. That is why a rule's weight wg cannot exceed the value 0.95.

Now let us calculate weight wg for the exemplary rules $r3$ and $r4$. First, we calculate lengths $l_{0.95,r3}$ and $l_{0.95,r4}$ of the confidence intervals for factors irf_{r3} and irf_{r4} , respectively. They are as follows: $l_{0.95,r3} = 0.13$; $l_{0.95,r4} = 0.04$. It means that the exactness of rule $r4$: $1 - l_{0.95,r4} = 0.96$ is higher than the exactness of rule $r3$: $1 - l_{0.95,r3} = 0.87$. It happens even though the number $N = 231$ (significant for rule exactness) is the same in the both rules. Here it is crucial that the conclusion of rule $r4$ is more 'characteristic' than the conclusion of rule $r3$.

Then, by means of formula (8), we determine rules' weights:

$$wg_{r3} = \min\{0.95, 0.87\} = 0.87 \quad \text{and} \quad wg_{r4} = \min\{0.95, 0.96\} = 0.95.$$

This means that $r4$ has a higher weight compared to $r3$. It is worth noticing that in case of rule $r3$, the exactness of interval estimation is crucial for its weight, whereas in case of rule $r4$, the confidence level of estimation $1 - \alpha = 0.95$ is crucial for its weight. In the latter case, an increase of the confidence level of estimation would cause an improvement of the rule's weight. The influence of this parameter on the rule's weight will be the subject of our future investigation.

4.2 Rule's Accuracy

Now we will discuss the second parameter having, in our opinion, an influence on global rule reliability. It is a rule's accuracy. To determine its value, we must first estimate accuracies of all rule's premises. As it was emphasized in [14], along with the course of semantic data integration, there can be observed a decrease in the accuracy of a virtual data being designed. Parameter acf enabling to express the final fact-premise accuracy or final fact-conclusion accuracy is there defined as follows:

$$acf(F_k) = \begin{cases} \frac{\sum_{i=1}^m N_i}{N} \cdot \frac{|V_{ik}|}{|V_k|} & \text{for the disjunction} \\ \frac{\sum_{i=1}^m N_i}{N} \cdot \frac{|V_k|}{|V_{ik}|} & \text{for the conjunction} \end{cases} \tag{9}$$

where N stands for 'attribute_count' of all common attributes in virtual tuple T ; $|V_k|$ – for the cardinality of a countable set of values attached to an attribute corresponding to fact F_k in T ; N_i stands for 'attribute_count' of all common attributes in component

tuple T_i (for $1 \leq i \leq m$); $|V_{ik}|$ – for the cardinality of a countable set of values attached to an attribute corresponding to fact F_k in T_i .

Then, we can determine *accuracy* ac of the rule obtained from T , with z facts (being premises or conclusions) F_k (for $1 \leq k \leq z$), as the arithmetic mean of all these facts' accuracies:

$$ac = \frac{1}{z} \sum_{k=1}^z acf(F_k) \quad (10)$$

Parameter ac takes its value from the range $<0; 1>$. It can be high (close to 1) if all the facts, while integrating, decrease their accuracies by a few percent only. It takes the highest possible value 1 if none of the facts decreases its accuracy (cardinalities of matching set values must be equal in all partial tuples).

Because of the lack of place, we will not exemplify here the whole process of data integration and rule designing. We will rather focus our attention on comparing partial tuple T_1 and virtual tuple T (see Chapter 3). Let us notice that the set values attached to attributes *age_range* and *symptoms* are different in both tuples. The set value of attribute *age_range* from T_1 is a subset of its counterpart from T ($\{1, \dots, 7\} \oplus \subset \{1, \dots, 18\} \oplus$), and the set value of attribute *symptoms* in T_1 is a superset of its counterpart from T ($\{\text{coughing}, \text{wheezing}\} \odot \supset \{\text{coughing}\} \odot$). Let us assume that the accuracies of facts-premises from rules r_3 and r_4 (designed from virtual tuple T that was previously obtained by the integration of partial tuple T_1 with other similar partial tuples) are as follows:

$$\begin{aligned} \text{act}(\text{General_Diagnosis}=\{\text{pediatric_asthma}\} \odot) &= 0.93, & \text{act}(\text{symptoms}=\{\text{coughing}\} \odot) &= 0.93, \\ \text{act}(\text{Drug}=\{\text{short-act_beta2_agonist, inhaled_antichol}\} \odot) &= 1, & \text{act}(\text{age_range}=\{1, \dots, 18\} \oplus) &= 0.62 \end{aligned}$$

and the accuracy of their facts-conclusions are as follows:

$$\text{act}(\text{treatment_effects}=\{\text{no_hospital_admis}\} \odot) = 0.93 \text{ and } \text{act}(\text{adverse_effects}=\{\text{vomiting}\} \odot) = 1.$$

It means that the facts-premises *age_range* and *symptoms*, while integrating, decreased their accuracies by 38% and 7%, respectively. On the other hand, the fact *adverse_effects* didn't decrease its accuracy at all (the same set of values $\{\text{vomiting}\} \odot$ was attached to this attribute in all the partial tuples, including T_1). The assumptions allow to estimate accuracies of rules r_3 and r_4 from the formula (10):

$$ac_{r_3} = 0.88 \text{ and } ac_{r_4} = 0.9.$$

4.3 Calculation of Global Rules' Reliabilities

Finally, from (6), we can estimate global reliability factors of rules r_3 and r_4 :

$$grf_{r_3} = \min\{0.87, 0.88\} = 0.87 \text{ and } grf_{r_4} = \min\{0.95, 0.9\} = 0.9.$$

As we can see, factor grf_{r_4} is higher compared to factor grf_{r_3} . As a result, rule r_4 will have a higher priority in the designed RBS knowledge base than rule r_3 (see Sec. 4.4). In case of rule r_3 , the most significant parameter for determining grf_{r_3} was the rule's weight, whereas in case of rule r_4 , the most significant parameter for determining grf_{r_4} – the rule's accuracy.

To summarize, factor grf depends on a rule's weight and accuracy. When the process of data integration proceeds, the two parameters behave as follows: a rule's weight increases; and a rule's accuracy decreases with the progress of data integration. It means that, at the same time, data integration influences positively (a number of individuals 'caught' in the rule) and negatively (dispersion of the individuals' characteristics) on factor grf . In order to obtain a high grf value, very sophisticated techniques of data integration should be used.

4.4 Rule's Priority

To summarize, we determine that for each production rule (1) factor $\lfloor grf \cdot 100 \rfloor$ (the integral part of $grf \cdot 100$) will be the one deciding about a rule's priority. Obviously, the

higher grf , the higher priority of the rule. In case those factors are equal for two different rules from the agenda, we have to compare the reliabilities of facts-premises from these rules – the higher minimal irf of a rule's fact-premise ($\min\{p_1, p_2, \dots, p_n\}$ – see formula (2)), the higher priority of the rule. In the light of the above considerations, a high priority rule can be obtained only as a result of integration of huge amounts of data; it has characteristic conclusion and facts-premises of high accuracy and high reliability.

5 Conclusions

In the paper we have proposed a method for ranking rules with uncertainty obtained from true evidence base. The method can facilitate creating high quality diagnostic RBSs with uncertainty. The main task of such systems is making the greatest possible number of reliable hypotheses that match an analyzed object or situation, and – preferably – listing them in the order of their importance. The method helps to properly determine rules' priorities, which in turn – by influencing the order of firing rules – help to make all important hypotheses and estimate their certainties.

The proposed in the paper format of a rule with uncertainty is based on using reliability factors irf and grf . Each of the two represents a kind of conditional probability, coming from an implicit ignorance range $\langle p_{Bel}; p_{Pris} \rangle$. Global reliability factor grf depends on the number and mutual similarity of data from which the rule has been designed. It itself serves as a base for calculating the rule's priority. In turn, internal reliability factor irf is being applied for calculating the hypothesis' certainty. The calculation is being performed by means of a few functions for propagating uncertainty. The detailed definitions of the functions and theorems on their properties are out of scope of this work.

In the paper, only aggregate data have been considered. In the examples, they represent medical results of not individual patients but of groups of patients. However, the presented method of ranking rules can be easily adapted to individual data.

Acknowledgments. The research has been supported by PUT under grants DS 511-43-078-/2011 and DS 45-083/11 DS-2010, and by the Polish Ministry of Science and Higher Education under grant NP N516 369536.

References

1. Ligęza, A.: Logical Foundations for Rule-Based Systems, 2nd edn. Springer, Heidelberg (2006)
2. Bragaglia, S., Chesani, F., Mello, P., Sottara, D.: A Rule-Based Implementation of Fuzzy Tableau Reasoning. In: Dean, M., Hall, J., Rotolo, A., Tabet, S. (eds.) RuleML 2010. LNCS, vol. 6403, pp. 35–49. Springer, Heidelberg (2010)
3. Nalepa, G.J., Ligęza, A.: On ALSV Rules Formulation and Inference. In: Lane, H.C., Guesgen, H.W. (eds.) Proc. 2nd Int. Florida Artif. Intel. Res. Soc. Conf., pp. 396–401. AAAI Press, Florida (2009)
4. Jankowska, B.: Using Semantic Data Integration to Create Reliable Rule-based Systems with Uncertainty. Eng. Appl. Artif. Intel. (2011) doi: 10.1016/j.engappai.2011.02.013
5. Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Mathematics in Science and Engineering, vol. 144. Academic Press, New York (1980)

6. Shafer, G.: *A Mathematical Theory of Evidence*. Princeton University Press, Princeton (1976)
7. Buchanan, B.G., Shortliffe, H. (eds.): *Rule-Based Expert Systems. The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading (1984)
8. Beeri, C., Ramakrishnan: *On the Power of Magic*. *J. Logic Program* 10, 255–299 (1991)
9. Agraval, R., Imielinski, T., Swani, A.: *Mining Association Rules Between Sets of Items in Large Databases*. *SIGMOD Rec.* 22(2), 805–810 (1993)
10. Van der Gaag, L.C.: *A Conceptual Model for Inexact Reasoning in Rule-Based Systems*. *Int. J. Approx. Reason.* 3(3), 239–258 (1989)
11. Szymkowiak, M., Jankowska, B.: *Discovering Medical Knowledge from Data in Patients' Files*. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) *ICCCI 2009. LNCS(LNAI)*, vol. 5796, pp. 128–139. Springer, Heidelberg (2009)
12. Plotnick, L.H., Ducharme, F.M.: *Combined Inhaled Anticholinergics and Beta2-Agonists for Initial Treatment of Acute Asthma in Children*. *The Cochrane Library* (2005)
13. Krysicki, W., et al.: *Mathematical Statistics*. PWN, Warszawa (1994) (in Polish)
14. Szymkowiak, M., Jankowska, B.: *Reliability of Medical Production Rules Obtained by means of Aggregate Data Mining*. *Journal of Medical Informatics & Technologies* 14, 103–110 (2010)

Smart Work Workbench; Integrated Tool for IT Services Planning, Management, Execution and Evaluation

Mariusz Fraś, Adam Grzech, Krzysztof Juszczyszyn, Grzegorz Kołaczek,
Jan Kwiatkowski, Agnieszka Prusiewicz, Janusz Sobecki,
Paweł Świątek, and Adam Wasilewski

Wrocław University of Technology, Institute of Computer Science
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{Mariusz.Fras, Adam.Grzech, Krzysztof.Juszczyszyn,
Grzegorz.Kolaczek, Jan.Kwiatkowski, Agnieszka.Prusiewicz,
Janusz.Sobecki, Pawel.Swiatek, Adam.Wasilewski}@pwr.wroc.pl

Abstract. Smart Service Workbench (SSW) is an integrated tool devoted to support business processes in distributed IT environment based on Service Oriented Architecture (SOA) paradigm. The tool's scope of functionalities is divided into modules that cover the whole process starting from the arrival of service request and ending with a return results of request. The latter is assumed to be a service composed of well-defined components available in predefined and staging repository. The tool's modules are responsible for requirements analysis, services choice or service composition, communication and computational resources allocation in distributed environment as well as for resources utilization monitoring to services quality and security evaluation purposes.

Keywords: Service Oriented Architecture, users requirements analysis, service components, service composition, complex service planning and execution, distributed environment, quality of service, security.

1 Introduction

Systems based on SOA (*Service Oriented Architecture*) paradigm offer services (complex services) which are delivered as composition of atomic services [11,12]. The main feature of such an attempt is that the required complex services may be efficiently and flexibly composed of available atomic services providing certain, well defined, required and personalized functionality. Requested complex services are characterized by set of parameters specifying both functional and nonfunctional requirements; the former define exact data processing procedures, while the latter describe various aspects of required service quality. The set of parameters describing requested complex service form SLA (*Service Level Agreement*) [1,15].

Functionality of the requested complex service is available as a sum of components functionalities. In order to deliver complex service with requested functional and non-functional properties appropriate components must be chosen in the process of services selection or composition [10]. Required functionality, uniquely defined in the

SLA, determines set of required atomic services as well as a plan according which the atomic services are performed in distributed environment. Non-functionality of the requested complex service, which is mainly related to QoS (*Quality of Service*) in distributed environment, may be assured or obtained by proper resources (processing and communication) and tasks (atomic services) allocation [6,7].

Presented tool, devoted to select required services, to compose required services, to allocate communication and computational resources, to control services execution, to compare required and available services, to evaluate quality of delivered services as well as to measure quality of mapping and services executed in distributed environment [3-5].

2 Smart Service Workbench Framework

Smart Service Workbench (SSW) is a framework composed of interacting modules that offer complex processing of service requests (Figure 1).

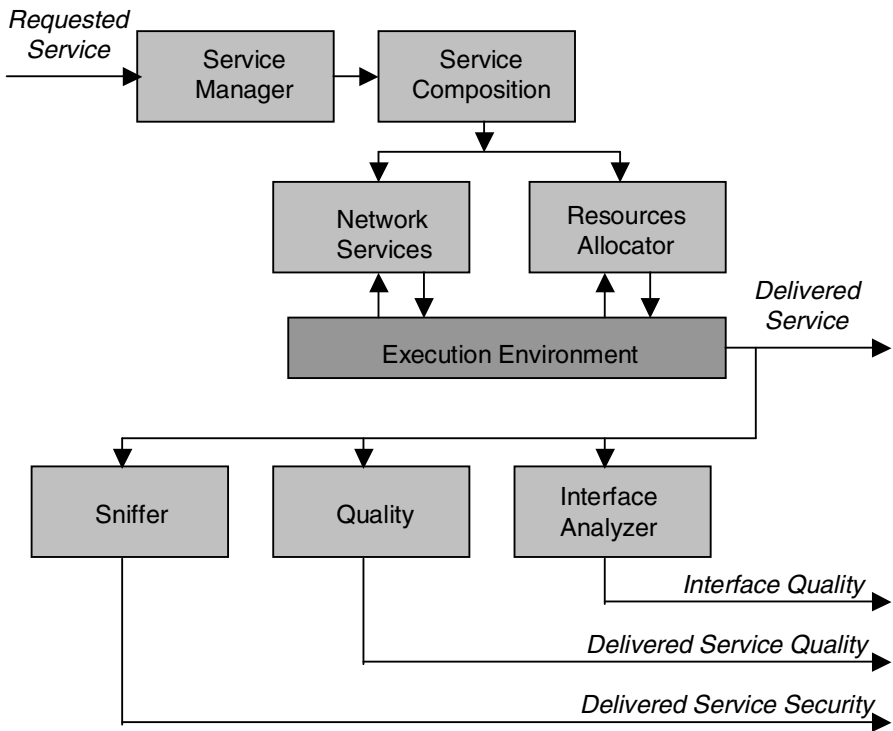


Fig 1. Smart Service Workbench framework overview

The framework covers service request processing starting from a service request arrival till complete the request and evaluation of the delivered service. The main functionality of SSW is delivered by Service Manager (request processing to select proper service matched to the request or to suggest service composition as a response for the service request), Service Composition (composition of service form components available at the system in gain to assure requested functionalities and non-functionalities), Network Services (broker of communication services in distributed environment) and Resources Allocator (broker of processing services in virtualized, distributed environment). The above listed are supported by modules: Attribute Analyzer (processes parameters describing users and services to evaluate their suitability to differentiate and discover common activities) and Networks (collects and processes knowledge about users behavior and services usage). Delivered services are analyzed at modules: Quality (evaluates values of parameters characterizing quality of delivered services), Sniffer (detects anomalies in users behavior and services usage) and Interface Analyzer (evaluates usability of service interfaces and suggests desired changes) in gain to compare requested and delivered services. Additionally, the framework contains Network Simulator, supporting Network Services module, in gain to examine various quality of services strategies in distributed environment testing purposes.

All the modules of the considered tool are design and implement as an open, modular, scalable and heterogeneous components integrated according SOA-based concept. Such an attempt allows to integrate all or selected modules to support various decision making processes. All the mentioned modules can be applied as independent data processing units or as a sequences of units supporting selected parts of requirements and services matching processes for given business process supported by SOA-based or legacy information systems.

In the subsequent parts of the work the above previously mentioned modules' architecture and functionalities are characterized.

2.1 Service Manager

Service Manager is a tool for requested service personalization and user access to web services environment management. It allows to analyze data about user behavior such as determining the characteristics of aggregates - user profiles, including extraction of patterns of activities, profiles of services. Knowledge, collected about the users and their activities within the IT system, is applied to recommend and personalize services. The Service Manager module is composed of two functional blocks: Profile Manager and Service Access (Figure 2).

Profile Manager is composed of blocks delivering data clustering, models and data classification, behavioral and service execution patterns. The main goal of data processing is extraction of patterns in datasets to support decision making and analysis processes. The functionality hidden in data clustering block is responsible for grouping users according to various criteria specified in clustering request. Using building models of classifiers block it is possible to build models of classifiers using past observations, which contains vector of features values of the objects and corresponding class labels. Built models of classifiers are used to classify according to vectors of features values. Profile Manager has ability to select automatically classification model by analyzing description of the data given in the classification

request. User can also chose classifiers from the list, which contains classifiers satisfying individual conditions about classification accuracy. Blocks of behavioral patters extraction are responsible for determining aggregated characteristics of services usage, which contains statistical, temporal and sequence patterns.

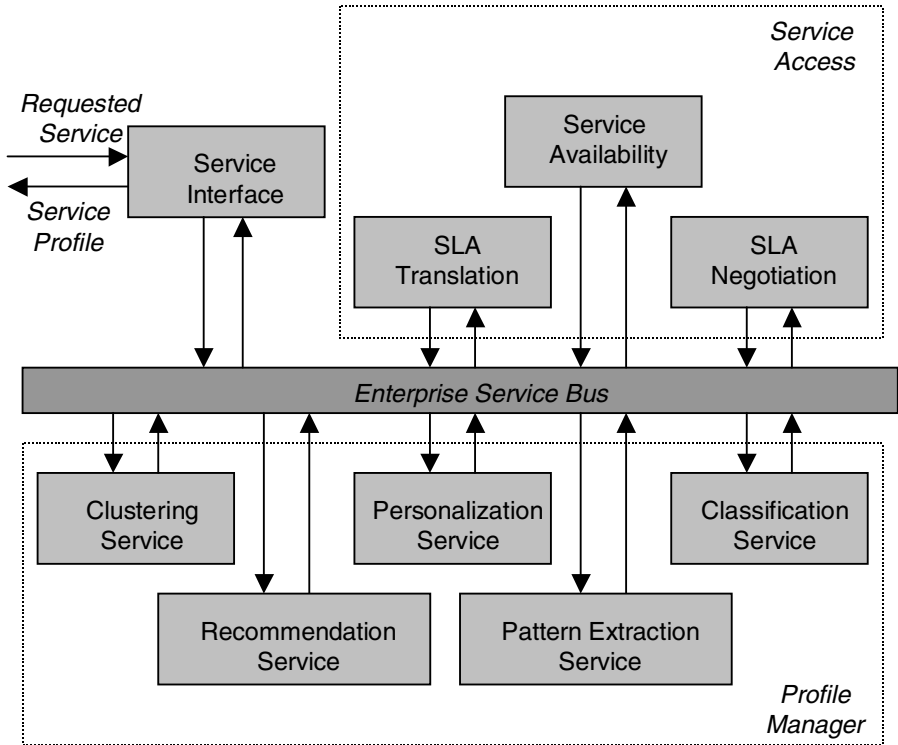


Fig. 2. Service Manager building blocks

Service Access is composed of blocks devoted to user's request transformation, profile-based estimation of ability to deliver requested service, SLA negotiation and recommendation as well as offered service profile [16].

Incoming service requests are translated in gain to express them in terms of formal ontology assumed for automatic processing of requests. Translated request is processed to obtain profile-based estimation of service execution ability. In result, three decisions are possible: request is rejected (requested service is unavailable at all), request is accepted (requested service is available in the services repository or as exactly as desired or similar to the desired), or request is transferred to composition.

The considered module is composed of blocks assuring objects (users, services, services usage patterns) clustering, classification and pattern recognition which are integrated via Enterprise Service Bus (ESB).

The Service Manager output, i.e., service profile being a service execution plan, is passed to resources broker (to perform selected service) or to service composition (to compose required service) modules.

2.2 Service Composition

Service Composition is a tool which gain is to propose service as specified by service profile being an output of Service Access module. The service is composed on components available in services repository taking into account requested functional and nonfunctional requirements as well as security assessments [12].

Service Composition module obtains service profile form the Service Manager module and is supported by Networks module (Figure 3).

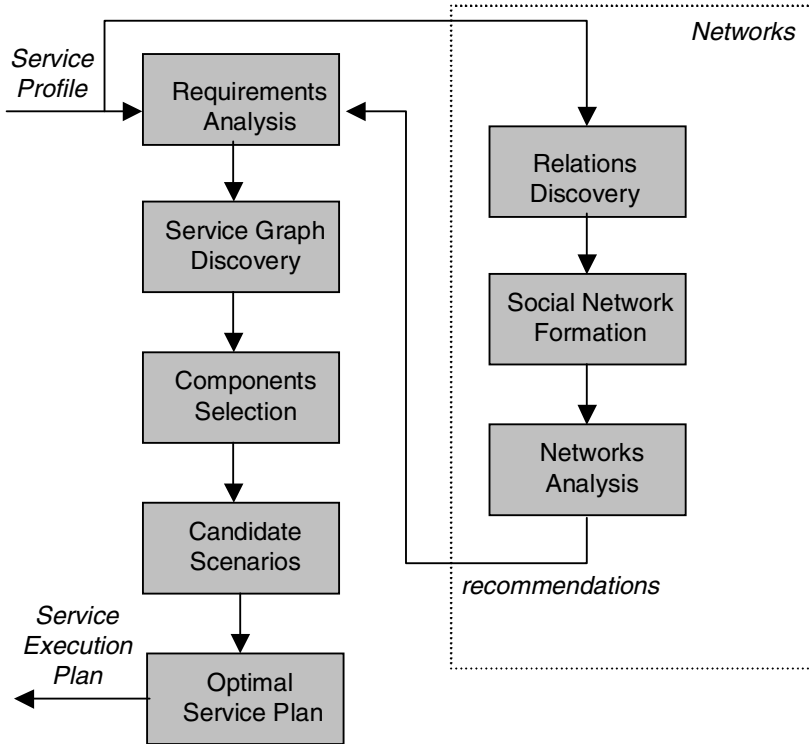


Fig. 3. Service Composition and Networks modules scheme

Composition of new services is carried on the basis of functional requirements (default), non-functional requirements (optional) and security requirements (optional).

The service composition process is considered as a three stage process: functional requirements graph (service graph) discovery, subsets of candidate components selection and components choice based on assumed criteria. The three mentioned stages may be performed in sequence or individually as depend on request complexity as well as on request and components ontology matching. On the basis of received service profile request the graph of functional requirements for its components is created. This graph is then used to generate the candidate scenarios of the required

service. For each scenario, the components are chosen from repository with respect to their functionalities (which must semantically conform to the requirements of the scenario). If required, the security assessment for the complex service is carried on. The last step is a multi-criterion selection of the optimal plan of the required complex service. The requested service execution plan is passed to the resources (communication and computational) broker modules.

Service Composition module obtains requested service profile from the Service Manager module and is supported by Networks module (Figure 1). The latter offers functionalities aiming to support the service composition processes. It tracks the requested service profile and previously generated execution plans in order to discover the relations (temporal, functional, communicational) between the users and required services. The discovered relations are then used to note the relation network which allows to group users, pick the characteristic ones, or predict their actions and services consumption.

Output of the Service Composition module, i.e., service execution plan is transferred to communication and computational resources brokers (working on parallel or in sequence). The Service Composition module architecture is SOA based; main building blocks are connected via ESB and the service composition process is performed by exchanging requests among services repository and service composer.

2.3 Network Services

Network Services is a broker of communication resources in given distributed environment. This broker aims is to recommend computational resources required to perform required service (defined as a service execution plan) based on on-line or off-line estimation of communication costs within distributed environment (Figure 4).

Functionality of the discussed broker is determined by scope of tasks that should be solved just after a new arriving service profile (from Service Manager module) or service execution plan (from Service Composition module) requires to perform various services components copies of which are kept in many execution system's nodes. Moreover, it is assumed that the number and localization of service components may change in time as depending of the volume of requests and spatial distribution of requests origins.

The broker collects and processes data about various components instances (components versions) localization, availability and occupancy in loosely or in more tightly coupled systems in gain to be ready indicate where the new arriving service request should be performed according required QoS strategy. Analysis of components' nonfunctional requirements gain is to limit set of available components copies and localizations to satisfy requirements.

The broker estimates service transfer times and service response time with the aid of adaptive services and communication links models, both built for every instance of service components. The models are designed as fuzzy-neural controllers (in the form of fuzzy-neural networks). Based on updated network model the broker selects localizations and communication paths to the selected localization where the component is performed. The broker also coordinates service components execution according assumed or given schedule [12].

Moreover, the module monitors resources utilization and quality of communication services; the monitored values are used to adapt (learn) models of services and models of links to various computational resources. These models are then used to calculate various service statistics that are recorded in broker repository.

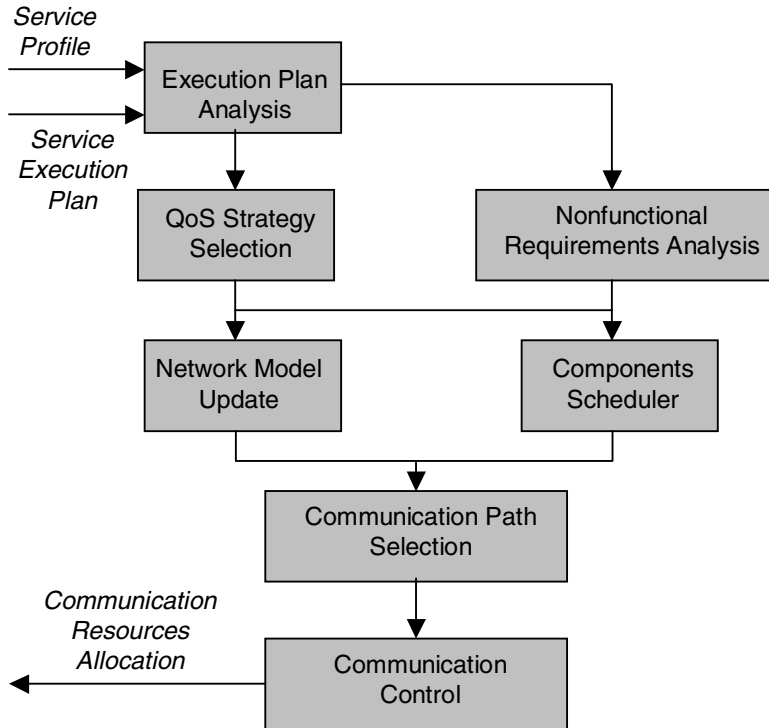


Fig. 4. Network Service scheme functional blocks

The discussed communication resources allocation module cooperates with computational resources allocator and the gain of the cooperation is to find proper resources allowing to perform the required service optimally (sub-optimally).

2.4 Computational Resources Allocator

The module is a tool for allocating computing resources to implement the service component. To indicate the computational resources, knowledge about the allocated communication resources and the current loading of computing resources are used. It allows to apply various computational resources, dynamically matched to satisfy the requirements in virtual environment. The decision on the allocation of computing resources is then confronted with the utilization of resources allocated using data from the service components execution monitoring. The latter is to collect knowledge about

the quality of services performed in particular environment, and to validate quality of methods applied to allocate resources (Figure 5).

The main gain of the module is to indicate the available computational resources based on updated knowledge about the current load of computational resources and to creation and use of available services irrespective of the hardware architecture and ensuring the efficient use of hardware resources.

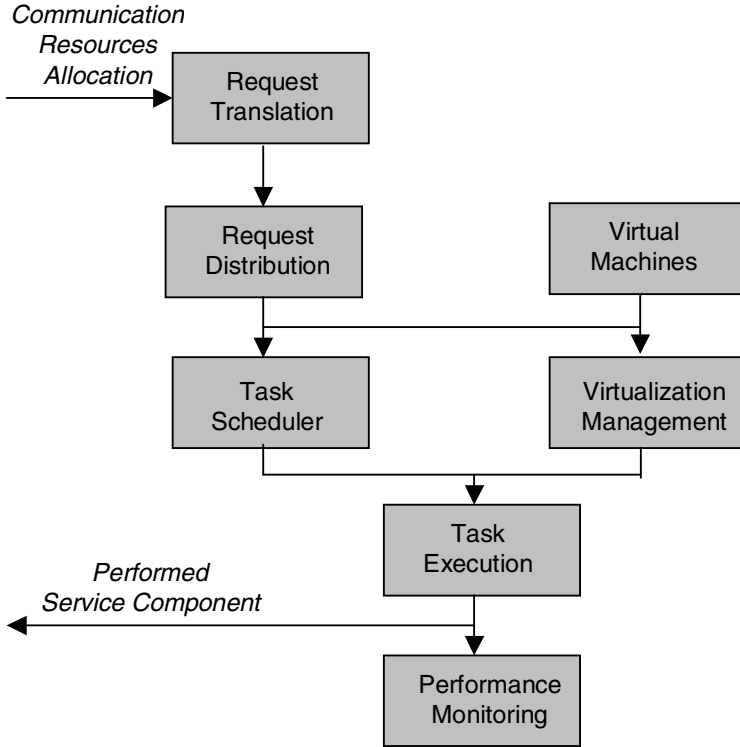


Fig. 5. Computational Resources Allocator components

The considered module architecture fulfils general concepts of SOA paradigm and resources virtualization in services composition and services execution in distribute environment. Compliance with the above concepts assures that the services delivery is independent on available hardware architecture and load balancing strategies may be applied. The latter ensures efficient and flexible usage of hardware resources in heavy load conditions [14].

2.5 Sniffer

The Sniffer module a tool for monitoring the status of the system’s security and identification of risk factors which can be used for improvement of security management of the organization (Figure 6)..

The discussed module offers two classes of services. The first one is a monitoring service which examines selected characteristics of the system (e.g., patterns of communication between the services, the intensity of the communication, the time of implementation services, etc.) The second class of services is to analyze the results obtained from the monitoring service to detect and to identify security incidents in heterogeneous and distributed service-oriented systems. The considered module provides its functionality using dedicated software agents. There are three classes of agents: agents specialized in detection of the global anomalies in the system's behavior, agents specialized in an anomaly detection of the separate service behavior, and the agents which detect anomalies in a wireless clients' behavior [12].

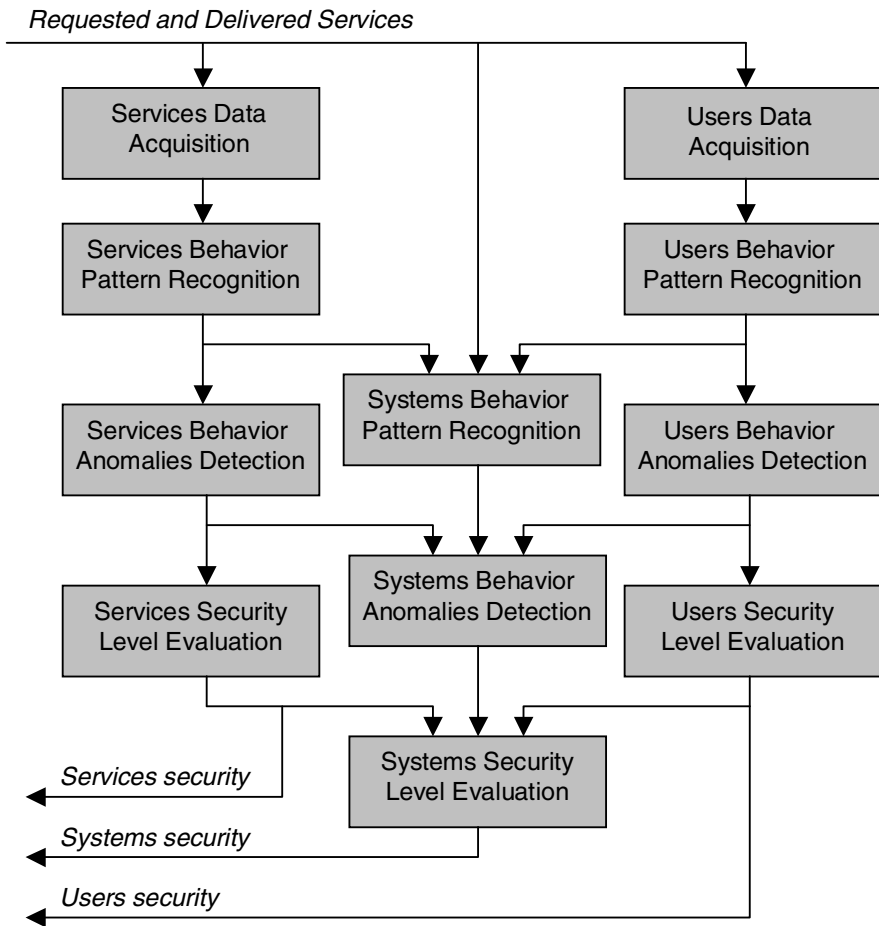


Fig. 6. Sniffer module three data processing tracks

Designed multi-agent architecture introduces the following types of agents: service security level monitoring agents, system security level monitoring agents and client behavior security level monitoring agents. In the first layer – service security level - a

collection of specialized agents which are directly responsible for the monitoring and preliminary processing of data derived from the service-oriented system. At the second layer – the agents responsible for the evaluation of security level of the system - the data from the agents from the lower layer are aggregated. At the third layer - users security level - data provided by other two agents layers are aggregated to discover user behavior patterns so to be able to evaluate security level, and further to detect their abnormal behavior.

The task carried out by a block analysis of the security requirements in the profile of IT service requests performs comparison of the security level requirements defined by the user and saved in the service request. The effect of the evaluation is the numeric value representing the level of the discrepancies between the desired by user security level and security level defined in the request for a service. The Security analysis of a complex service execution plan block shall evaluate the level of security available for a particular execution plan of a complex service. The block of security level analysis of services after the allocation of computing and communication resources evaluates the security of the service-oriented system using in this analysis all available information about computing and communication resources consumption level within the system being monitored. The last block is responsible for the assessment of the accordance level of the expected by the user security level and the actual one.

The considered module is composed of users, services and system monitoring agents integrated with decision making components (data acquisition, pattern recognition, anomalies detection and security levels estimation) via Enterprise Service Bus (ESB) with proper users interface.

2.6 Quality

Quality module supports quality management in service-oriented systems (Figure 7).

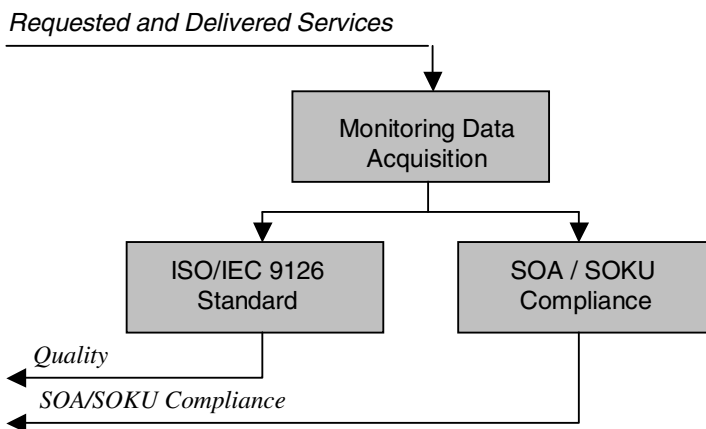


Fig. 7. Quality module components

The starting point of quality analysis are the needs and requirements of users, based on which the quality factors values (and results interpretation), covering different aspects of service-oriented systems are determined. The tool supports i.e. evaluating the effectiveness of the application running time (time behavior), quality perceived by users (based on questionnaires which is an adaptation of well-known questionnaire), as well as services compose ability, granularity, flexibility, heterogeneity, distribute ability, scalability, reusability, etc.

The considered module supports quality estimation and analysis using selected quality characteristics described by the ISO 9126 standard (Product Software Quality) and the characteristics of the model of compliance with the SOA/SOKU (Service Oriented Knowledge Utilities), which is an integral part of the tool. The latter functionality is devoted to characterize and measure level at which the knowledge, collected during the service composition and perform processes, is utilized [13].

2.7 Interface Analyzer

Interface Analyzer module is a tool, combining of methodology and algorithms, devoted to analyze quality of systems interfaces and, if required, to recommend changes of the interface in terms of interface usability (Figure 8).

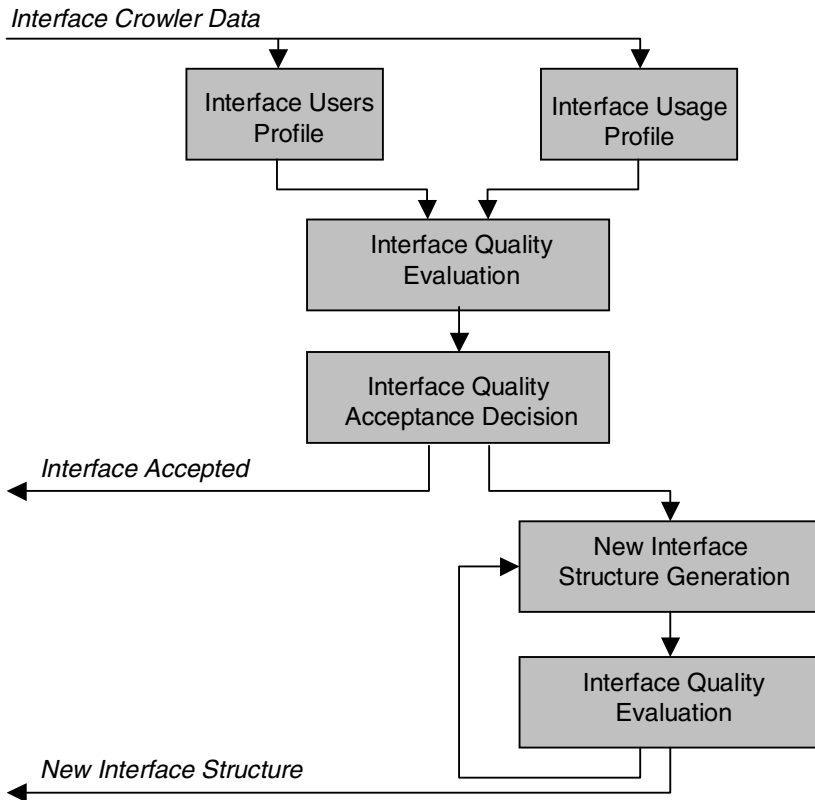


Fig. 8. Interface Analyzer functional blocks

The main idea of the system interface analysis is based on assumption that the usage of particular system interface by particular user may be monitored and processed in gain to evaluate the interface quality. If the latter is below some accepted quality level some changes in the interface may be recommended. The web system interface quality evaluation method, implemented in the discussed module, allows to process both static and dynamic interface usage patterns. Moreover, it is useful to analyze the personalized and in average usage of the interfaces.

The evaluation of the system interface starts when the interface structure is discovered, i.e., if so called interface usage graph is built. Observation and measures of the interface usage by particular user or group of users allow to collect data in gain to update values of parameters of the usage graph. The usage graph's nodes and arc are characterized by various parameters required to determine aggregate value of parameter, called graph energy, which represents interface usability. Until the graph's energy is below some assumed level (given by experts or calculated for interfaces considered as acceptable) the decision making blocks recommends to keep the interface as it is.

If the quality level of the interface does not satisfy requirements the decision making block recommends interface changes. Changes of the interface means changes in the interface structure, i.e., number of nodes and arcs among nodes in the usage graph. Collected and kept data about the interface usage are useful in evaluation of new recommended interface structures.

2.8 Additional Modules

The above presented set of tools is supported by two additional tools: Attributes Analyzes and Network Simulator.

Attribute Visualizer. The Attribute Visualizer is useful for multidimensional data analysis and visualization purposes by means of data set dimension reduction (Figure 9).

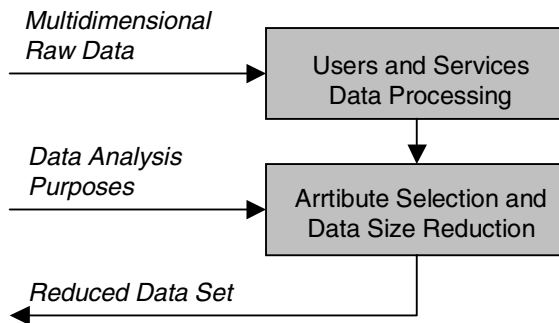


Fig. 9. Attribute Visualizer main components

In general it enables selection of useful attributes (about users and services) by means of rough classification method. The gain of dimension reduction is to determine minimal set of attributes for high quality prediction of services quality evaluation (users and services grouping, identification, pattern recognition, etc.), i.e., to specify optimal list of attributes for user, service and service usage profiles analysis purposes [2,3].

The basic element of the considered tool is a numerical optimization algorithm that iteratively distinguishes the optimal representation of the set of objects described by multidimensional data. The result of each iteration is a better approximation of the expected result. Iterative procedure enables control of the accuracy of the results regarding available computational resources. Selection of the useful attributes by means of rough classification procedure enables to select optimal subset of attributes for prediction purposes and from the service utility point of view. Attributes size reduction limits both services evaluation computational complexity and amount of sensitive data processed for decision making purposes.

Network Simulator. The module is a tool to research and test the algorithms for the distribution of requests in the service-oriented environment. It is devoted to model and simulate events in a service-oriented system. Wide scope of the simulator functionality for event-driven simulation is assured by the simulator main building blocks: service requests generator, service components creator, communication and computational services broker connected via services bus.

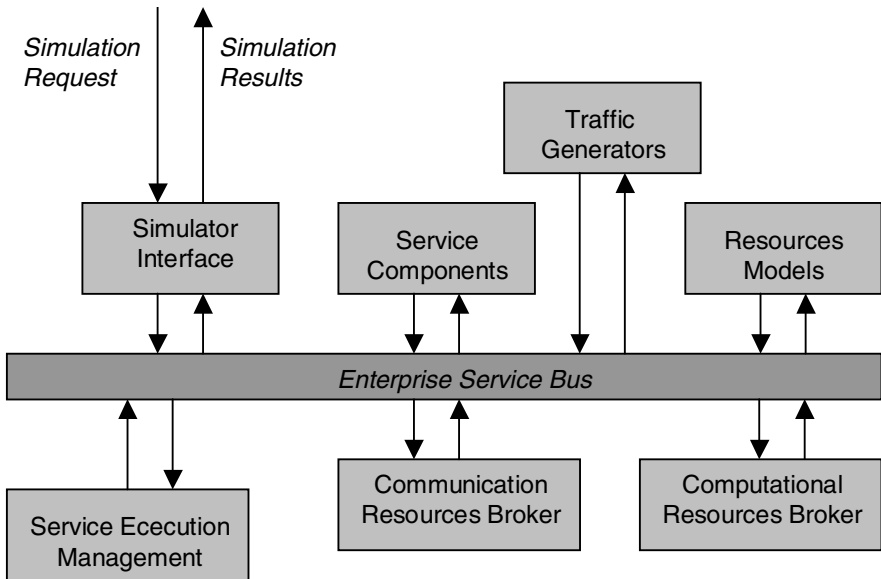


Fig. 10. Network Simulator scheme

The considered simulator of service-oriented system offers simulation of a service-oriented system with focus on requests distribution algorithms, statistical analysis of simulation results, easy implementation of reconfigurable system models, implementation of various quality of service strategies and simulation in batch and graphics mode [6-9].

During the implementation of the discussed simulation environment, the emphasis has been put on easy implementation of algorithms, distribution of requests in the

broker and the availability of statistics to analyze the behavior of the algorithm. Simulations are configurable via a configuration file that allows, *inter alia*, to run a batch simulations for algorithm performance analysis for a given list of parameters values. One can also run the simulation in graphics mode with an animations for better understanding of the simulated process.

3 Conclusion

The SSW set of tools, presented above, is designed in gain to support decision making tasks which may be distinguished in the process of matching required (business process) and available (computer-based information system) services.

The business process and IT systems' services integration scenario as well as required and available services matching process is divided into several, well-defined steps which covers all activities that should be undertaken between request arriving time till service delivery time (end-to-end working prototype). The scope of functionalities obtainable at distinguished steps may be easily extended by adding new procedures, methods and algorithms.

The general concepts, implemented in the set of modules, is based on component oriented software development idea; required business process information services may be delivered performing predefined (available form services repository) or composed on demand from service components (service on demand) distributed IT systems services. The presented tool is an attempt to offer various functionalities delivered as a services: data processing as a service, security as a service, composition as a service, infrastructure as a service, monitoring as a service, etc. Such an assumption means also that functionalities available in the above presented modules are reusable at different steps in the services matching process.

Functionality of the presented modules are based on extensive data, information and knowledge collecting and processing. It is evident especially in service manager and service composition modules where ontology matching and prediction algorithm are used to select proper services or to obtain optimal services composition of components available in various instances in space-distributed environment.

The presented tool is still under development. The general framework idea assures that all presented modules functionalities may be easily extended by adding general-purpose and specific-oriented data, information and knowledge collecting and processing units. Moreover, it is assumed that both available and planned processing capabilities are reusable both in many parallel services delivery processes as well in various steps of the same distinguished service delivery process.

The functionality of the describe tool presents also multi-steps methodology for semantically annotating services matching, selection, composition and execution. There are several innovations in the presented attempt. First of all, the services matching (management) is available in three versions: service selection, service level agreement negotiation and service composition. The second is that the services composition (service on demand) process is decoupled into four stages that allows to utilize various kinds of knowledge to obtain optimal composed service.

Acknowledgment. The research presented in this paper has been partially supported by the European Union within the European Regional Development Fund program no. POIG.01.03.01-00-008/08.

References

1. Anderson, S., Grau, A., Hughes, C.: Specification and satisfaction of SLAs in service oriented architectures. In: 5th Annual DIRC Research Conference, pp. 141–150 (2005)
2. Brzostowski, K., Rekuć, W., Sobiecki, J., Szczurowski, L.: Service Discovery in the SOA System. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) *Intelligent Information and Database Systems*. LNCS(LNAI), vol. 5991, pp. 29–38. Springer, Heidelberg (2010)
3. Drapała, J., Żatuchin, D., Sobiecki, J.: Multidimensional data visualization applied for user's questionnaire data quality assessment. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) *KES-AMSTA 2010*. LNCS, vol. 6070, pp. 351–360. Springer, Heidelberg (2010)
4. Garey, M., Johnson, D., Sethi, R.: The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research* 1, 117–129 (1976)
5. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics* 3, 287–326 (1979)
6. Grzech, A., Świątek, P.: Parallel processing of connection streams in nodes of packet-switched computer communication networks. *Cybernetics and Systems* 39(2), 155–170 (2008)
7. Grzech, A., Świątek, P.: Modeling and optimization of complex services in service-based systems. *Cybernetics and Systems* 40(8), 706–723 (2009)
8. Grzech, A.: Resources utilization in distributed environment for complex services. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) *Intelligent Information and Database Systems*. LNCS, vol. 5991, pp. 400–409. Springer, Heidelberg (2010)
9. Grzech, A., Rygielski, P., Świątek, P.: Dynamic resources allocation for delivery of personalized services. In: Cellary, W., Estevez, E. (eds.) *Software Services for e-World*. IFIP Advances in Information and Communication Technology, vol. 341, pp. 17–28. Springer, Heidelberg (2010)
10. Jaeger, M.C., Rojcek-Goldmann, G., Muhl, G.: QoS aggregation in web service compositions. In: *IEEE Int. Conf. on e-Technology, e-Commerce and e-Service*, pp. 181–185 (2005)
11. Johnson, R., Gamma, E., Helm, R., Vlisides, J.: *Design patterns; elements of reusable object-oriented software*. Addison-Wesley, Reading (1995)
12. Kolaczek, G., Juszczyszyn, K.: Smart Security Assessment of Composed Web Services. *Cybernetics and Systems* 41(1), 46–61 (2010)
13. Kolaczek, G., Wasilewski, A.: Software Security in the Model for Service Oriented Architecture Quality. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) *PPAM 2009*. LNCS, vol. 6067, pp. 226–235. Springer, Heidelberg (2010)
14. Kwiatkowski, J., Fraś, M., Pawlik, M., Konieczny, D.: Request distribution in hybrid processing environments. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) *PPAM 2009*. LNCS, vol. 6067, pp. 246–255. Springer, Heidelberg (2010)
15. Milanovic, N., Malek, M.: Current Solutions for Web Service Composition. *IEEE Internet Computing* 8(6), 51–59 (2004)
16. Prusiewicz, A., Zięba, M.: Services Recommendation in Systems Based on Service Oriented Architecture by Applying Modified ROCK Algorithm. *CCIS*, pp. 226–238. Springer, Prague

A Cut-Free ExpTime Tableau Decision Procedure for the Description Logic SHI

Linh Anh Nguyen

Institute of Informatics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
nguyen@mimuw.edu.pl

Abstract. We give the first cut-free EXPTIME (optimal) tableau decision procedure for checking satisfiability of a knowledge base in the description logic *SHI*, which extends the description logic *ALC* with transitive roles, inverse roles and role hierarchies.

1 Introduction

Ontologies provide a shared understanding of the domain for different applications that want to communicate to each other. They are useful for several important areas like knowledge representation, software integration and Web applications. Web Ontology Language (OWL) is a layer of the Semantic Web architecture, built on the top of XML and RDF. Together with rule languages it serves as a main knowledge representation formalism for the Semantic Web. The logical foundation of OWL is based on description logics (DLs). Some of the most well-known DLs, in the increasing order of expressiveness, are *ALC*, *SH*, *SHI*, *SHIQ* and *SROIQ* [1,6].

Description logics represent the domain of interest in terms of concepts, individuals, and roles. A concept is interpreted as a set of individuals, while a role is interpreted as a binary relation among individuals. A knowledge base in a DL consists of axioms about roles (grouped into an RBox), terminology axioms (grouped into a TBox), and assertions about individuals (grouped into an ABox). One of the basic inference problems in DLs, which we denote by *Sat*, is to check satisfiability of a knowledge base. Other inference problems in DLs are usually reducible to this problem. For example, the problem of checking consistency of a concept w.r.t. an RBox and a TBox (further denoted by *Cons*) is linearly reducible to *Sat*.

In this paper we study automated reasoning in the DL *SHI*, which extends the DL *ALC* with transitive roles, inverse roles and role hierarchies. The aim is to develop an efficient tableau decision procedure for the *Sat* problem in *SHI*. It should be complexity-optimal (EXPTIME), cut-free, and extendable with useful optimizations. Tableau methods have widely been used for automated reasoning in modal and description logics [2] since they are natural and allow many optimizations. As *SHI* is a sublogic of *SROIQ* and REG^c (regular grammar logic with converse), one can use the tableau decision procedures of *SROIQ* [6]

and REG^c [14] for the *Sat* problem in \mathcal{SHI} . However, the first procedure has suboptimal complexity (NEXP_{TIME} when restricted to \mathcal{SHI}), and the second one uses analytic cuts.

The tableau decision procedure given in [7] for the *Cons* problem in \mathcal{SHI} has NEXP_{TIME} complexity.

In [3] together with Goré we gave the first EXP_{TIME} tableau decision procedure for the *Cons* problem in \mathcal{SHI} , which uses analytic cuts to deal with inverse roles. In [13] together with Szalas we gave the first direct EXP_{TIME} tableau decision procedure for the *Sat* problem in the DL \mathcal{SH} . In [9] we gave the first cut-free EXP_{TIME} tableau decision procedure for the *Sat* problem in the DL \mathcal{ALCI} .

In this paper, by extending the methods of [3,13,9], we give the first cut-free EXP_{TIME} (optimal) tableau decision procedure for the *Sat* problem in the DL \mathcal{SHI} . We use global state caching [4,5,9], the technique of [9] for dealing with inverse roles, the technique of [3,13] for dealing with transitive roles and hierarchies of roles, and the techniques of [13,12,14,9] for dealing with ABoxes.

The rest of this paper is structured as follows: In Section 2 we recall the notation and semantics of \mathcal{SHI} . In Section 3 we describe our tableau decision procedure for the *Sat* problem in \mathcal{SHI} . Section 4 concludes this work. Due to the lack of space, pseudocode of our decision procedure and proofs of our results are presented only in the long version [10] of the current paper.

2 Notation and Semantics of \mathcal{SHI}

Our language uses a finite set \mathbf{C} of *concept names*, a finite set \mathbf{R} of *role names*, and a finite set \mathbf{I} of *individual names*. We use letters like A and B for *concept names*, r and s for *role names*, and a and b for *individual names*. We refer to A and B also as *atomic concepts*, and to a and b as *individuals*.

For $r \in \mathbf{R}$, let r^- be a new symbol, called the *inverse* of r . Let $\mathbf{R}^- = \{r^- \mid r \in \mathbf{R}\}$ be the set of *inverse roles*. For $r \in \mathbf{R}$, define $(r^-)^- = r$. A *role* is any member of $\mathbf{R} \cup \mathbf{R}^-$. We use letters like R and S to denote roles.

An (\mathcal{SHI}) *RBox* \mathcal{R} is a finite set of role axioms of the form $R \sqsubseteq S$ or $R \circ R \sqsubseteq R$. By $ext(\mathcal{R})$ we denote the least extension of \mathcal{R} such that:

- $R \sqsubseteq R \in ext(\mathcal{R})$ for any role R
- if $R \sqsubseteq S \in ext(\mathcal{R})$ then $R^- \sqsubseteq S^- \in ext(\mathcal{R})$
- if $R \circ R \sqsubseteq R \in ext(\mathcal{R})$ then $R^- \circ R^- \sqsubseteq R^- \in ext(\mathcal{R})$
- if $R \sqsubseteq S \in ext(\mathcal{R})$ and $S \sqsubseteq T \in ext(\mathcal{R})$ then $R \sqsubseteq T \in ext(\mathcal{R})$.

By $R \sqsubseteq_{\mathcal{R}} S$ we mean $R \sqsubseteq S \in ext(\mathcal{R})$. If $R \sqsubseteq_{\mathcal{R}} S$ then R is a *subrole* of S w.r.t. \mathcal{R} . If $R \circ R \sqsubseteq_{\mathcal{R}} R$ then R is a *transitive role* w.r.t. \mathcal{R} .

Concepts in \mathcal{SHI} are formed using the following BNF grammar:

$$C, D ::= \top \mid \perp \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$$

We use letters like C and D to denote arbitrary concepts.

A *TBox* is a finite set of axioms of the form $C \sqsubseteq D$ or $C \doteq D$. An *ABox* is a finite set of *assertions* of the form $a : C$ (*concept assertion*) or $R(a, b)$ (*role*

assertion). A *knowledge base* in \mathcal{SHI} is a tuple $(\mathcal{R}, \mathcal{T}, \mathcal{A})$, where \mathcal{R} is an RBox, \mathcal{T} is a TBox and \mathcal{A} is an ABox.

A *formula* is defined to be either a concept or an ABox assertion. We use letters like φ, ψ and ξ to denote formulas, and letters like X, Y and Γ to denote sets of formulas.

An *interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ consists of a non-empty set $\Delta^{\mathcal{I}}$, called the *domain* of \mathcal{I} , and a function $\cdot^{\mathcal{I}}$, called the *interpretation function* of \mathcal{I} , that maps every concept name A to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, maps every role name r to a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, and maps every individual name a to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The interpretation function is extended to inverse roles and complex concepts as follows:

$$\begin{aligned} (r^-)^{\mathcal{I}} &= \{ \langle x, y \rangle \mid \langle y, x \rangle \in r^{\mathcal{I}} \} & \top^{\mathcal{I}} &= \Delta^{\mathcal{I}} & \perp^{\mathcal{I}} &= \emptyset \\ (\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} & (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} & (C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\ (\forall R.C)^{\mathcal{I}} &= \{ x \in \Delta^{\mathcal{I}} \mid \forall y [\langle x, y \rangle \in R^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}] \} \\ (\exists R.C)^{\mathcal{I}} &= \{ x \in \Delta^{\mathcal{I}} \mid \exists y [\langle x, y \rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}] \} \end{aligned}$$

Note that $(r^-)^{\mathcal{I}} = (r^{\mathcal{I}})^{-1}$ and this is compatible with $(r^-)^- = r$.

For a set Γ of concepts, define $\Gamma^{\mathcal{I}} = \{ x \in \Delta^{\mathcal{I}} \mid x \in C^{\mathcal{I}} \text{ for all } C \in \Gamma \}$.

The relational composition of binary relations R_1, R_2 is denoted by $R_1 \circ R_2$.

An interpretation \mathcal{I} is a *model of an RBox* \mathcal{R} if for every axiom $R \sqsubseteq S$ (resp. $R \circ R \sqsubseteq R$) of \mathcal{R} , we have that $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$ (resp. $R^{\mathcal{I}} \circ R^{\mathcal{I}} \subseteq R^{\mathcal{I}}$). Note that if \mathcal{I} is a model of \mathcal{R} then it is also a model of $\text{ext}(\mathcal{R})$.

An interpretation \mathcal{I} is a *model of a TBox* \mathcal{T} if for every axiom $C \sqsubseteq D$ (resp. $C \doteq D$) of \mathcal{T} , we have that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (resp. $C^{\mathcal{I}} = D^{\mathcal{I}}$).

An interpretation \mathcal{I} is a *model of an ABox* \mathcal{A} if for every assertion $a:C$ (resp. $R(a,b)$) of \mathcal{A} , we have that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ (resp. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$).

An interpretation \mathcal{I} is a *model of a knowledge base* $(\mathcal{R}, \mathcal{T}, \mathcal{A})$ if \mathcal{I} is a model of all \mathcal{R}, \mathcal{T} and \mathcal{A} . A knowledge base $(\mathcal{R}, \mathcal{T}, \mathcal{A})$ is *satisfiable* if it has a model.

An interpretation \mathcal{I} *satisfies* a concept C (resp. a set X of concepts) if $C^{\mathcal{I}} \neq \emptyset$ (resp. $X^{\mathcal{I}} \neq \emptyset$). A set X of concepts is *satisfiable w.r.t. an RBox* \mathcal{R} and a TBox \mathcal{T} if there exists a model of \mathcal{R} and \mathcal{T} that satisfies X . For $X = Y \cup \mathcal{A}$, where Y is a set of concepts and \mathcal{A} is an ABox, we say that X is *satisfiable w.r.t. an RBox* \mathcal{R} and a TBox \mathcal{T} if there exists a model of $(\mathcal{R}, \mathcal{T}, \mathcal{A})$ that satisfies X .

3 A Tableau Decision Procedure for \mathcal{SHI}

We assume that concepts and ABox assertions are represented in negation normal form (NNF), where \neg occurs only directly before atomic concepts.¹ We use \overline{C} to denote the NNF of $\neg C$, and for $\varphi = a:C$, we use $\overline{\varphi}$ to denote $a:\overline{C}$. For simplicity, we treat axioms of \mathcal{T} as concepts representing global assumptions: an axiom $C \sqsubseteq D$ is treated as $\overline{C} \sqcup D$, while an axiom $C \doteq D$ is treated as $(\overline{C} \sqcup D) \sqcap (\overline{D} \sqcup C)$. That is, we assume that \mathcal{T} consists of concepts in NNF.

¹ Every formula can be transformed to an equivalent formula in NNF.

Thus, an interpretation \mathcal{I} is a model of \mathcal{T} iff \mathcal{I} validates every concept $C \in \mathcal{T}$. As this way of handling the TBox is not efficient in practice, the absorption technique like the one discussed in [11,13] can be used to improve the performance of our algorithm.

From now on, let $(\mathcal{R}, \mathcal{T}, \mathcal{A})$ be a knowledge base in NNF of the logic \mathcal{SHI} , with $\mathcal{A} \neq \emptyset$.² In this section we present a tableau calculus for checking satisfiability of $(\mathcal{R}, \mathcal{T}, \mathcal{A})$. For a set X of concepts and a set Y of ABox assertions, we define:

$$\begin{aligned} \text{SRTR}_{\mathcal{R}}(R, S) &= (R \sqsubseteq_{\mathcal{R}} S \wedge S \circ S \sqsubseteq S \in \text{ext}(\mathcal{R})) \\ \text{Trans}_{\mathcal{R}}(X, R) &= \{D \mid \forall R.D \in X\} \cup \{\forall S.D \in X \mid \text{SRTR}_{\mathcal{R}}(R, S)\} \\ \text{Trans}_{\mathcal{R}}(X, R, a) &= \{a:D \mid \forall R.D \in X\} \cup \{a:\forall S.D \mid \forall S.D \in X \wedge \text{SRTR}_{\mathcal{R}}(R, S)\} \\ \text{Trans}_{\mathcal{R}}(Y, a, R) &= \{D \mid a:\forall R.D \in Y\} \cup \{\forall S.D \mid a:\forall S.D \in Y \wedge \text{SRTR}_{\mathcal{R}}(R, S)\} \\ \text{Trans}_{\mathcal{R}}(Y, a, R, b) &= \{b:D \mid a:\forall R.D \in Y\} \cup \\ &\quad \{b:\forall S.D \mid a:\forall S.D \in Y \wedge \text{SRTR}_{\mathcal{R}}(R, S)\} \end{aligned}$$

We call $\text{Trans}_{\mathcal{R}}(X, R)$ the *transfer of X through R w.r.t. \mathcal{R}* , call $\text{Trans}_{\mathcal{R}}(X, R, a)$ the *transfer of X through R to a w.r.t. \mathcal{R}* , call $\text{Trans}_{\mathcal{R}}(Y, a, R)$ the *transfer of Y starting from a through R w.r.t. \mathcal{R}* , and call $\text{Trans}_{\mathcal{R}}(Y, a, R, b)$ the *transfer of Y from a to b through R w.r.t. \mathcal{R}* .

In what follows we define tableaux as rooted “and-or” graphs. Such a graph is a tuple $G = (V, E, \nu)$, where V is a set of nodes, $E \subseteq V \times V$ is a set of edges, $\nu \in V$ is the root, and each node $v \in V$ has a number of attributes. If there is an edge $(v, w) \in E$ then we call v a *predecessor* of w , and call w a *successor* of v . The set of all attributes of v is called the *contents* of v . Attributes of tableau nodes are:

- $\text{Type}(v) \in \{\text{state}, \text{non-state}\}$. If $\text{Type}(v) = \text{state}$ then we call v a *state*, else we call v a *non-state* (or an *internal* node). If $\text{Type}(v) = \text{state}$ and $(v, w) \in E$ then $\text{Type}(w) = \text{non-state}$.
- $\text{SType}(v) \in \{\text{complex}, \text{simple}\}$ is called the subtype of v . If $\text{SType}(v) = \text{complex}$ then we call v a *complex node*, else we call v a *simple node*. The graph never contains edges from a simple node to a complex node. If (v, w) is an edge from a complex node v to a simple node w then $\text{Type}(v) = \text{state}$ and $\text{Type}(w) = \text{non-state}$. The root of the graph is a complex node.
- $\text{Status}(v) \in \{\text{unexpanded}, \text{expanded}, \text{incomplete}, \text{unsat}, \text{sat}\}$.
- $\text{Label}(v)$ is a finite set of formulas, called the label of v . The label of a complex node consists of ABox assertions, while the label of a simple node consists of concepts.
- $\text{RFmls}(v)$ is a finite set of formulas, called the set of reduced formulas of v .
- $\text{DFmls}(v)$ is a finite set of formulas, called the set of disallowed formulas of v .
- $\text{StatePred}(v) \in V \cup \{\text{null}\}$ is called the state-predecessor of v . It is available only when $\text{Type}(v) = \text{non-state}$. If v is a non-state and G has no paths connecting a state to v then $\text{StatePred}(v) = \text{null}$. Otherwise, G has exactly

² If \mathcal{A} is empty, we can add $a:\top$ to it, where a is a special individual.

one state u that is connected to v via a path not containing any other states. In that case, $StatePred(v) = u$.

- $ATPred(v) \in V$ is called the after-transition-predecessor of v . It is available only when $Type(v) = \text{non-state}$. If v is a non-state and $v_0 = StatePred(v)$ ($\neq \text{null}$) then there is exactly one successor v_1 of v_0 such that every path connecting v_0 to v must go through v_1 , and we have that $ATPred(v) = v_1$. We define $AfterTrans(v) = (ATPred(v) = v)$. If $AfterTrans(v)$ holds then either v has no predecessors (i.e. it is the root of the graph) or it has exactly one predecessor u and u is a state.
 - $CELabel(v)$ is a formula called the coming edge label of v . It is available only when v is a successor of a state u (and $Type(v) = \text{non-state}$). In that case, we have $u = StatePred(v)$, $AfterTrans(v)$ holds, $CELabel(v) \in Label(u)$, and
 - if $SType(u) = \text{simple}$ then

$$CELabel(v) \text{ is of the form } \exists R.C \text{ and } C \in Label(v)$$
 - else $CELabel(v)$ is of the form $a:\exists R.C$ and $C \in Label(v)$.
- Informally, v was created from u to realize the formula $CELabel(v)$ at u .
- $ConvMethod(v) \in \{0, 1\}$ is called the converse method of v . It is available only when $Type(v) = \text{state}$.
 - $FmlsRC(v)$ is a set of formulas, called the set of formulas required by converse for v . It is available only when $Type(v) = \text{state}$ and will be used only when $ConvMethod(v) = 0$.
 - $AltFmlSetsSC(v)$ is a set of sets of formulas, called the set of alternative sets of formulas suggested by converse for v . It is available only when $Type(v) = \text{state}$ and will be used only when $ConvMethod(v) = 1$.
 - $AltFmlSetsSCP(v)$ is a set of sets of formulas, called the set of alternative sets of formulas suggested by converse for the predecessor of v . It is available only when v has a predecessor being a state and will be used only when $ConvMethod(v) = 1$.

We define

$$\begin{aligned} AFmls(v) &= Label(v) \cup RFmls(v) \\ NDFmls(v) &= \{\bar{\varphi} \mid \varphi \in DFmls(v)\} \\ FullLabel(v) &= AFmls(v) \cup NDFmls(v) \\ Kind(v) &= \begin{cases} \text{and-node} & \text{if } Type(v) = \text{state} \\ \text{or-node} & \text{if } Type(v) = \text{non-state} \end{cases} \\ BeforeFormingState(v) &= v \text{ has a successor which is a state} \end{aligned}$$

The sets $AFmls(v)$, $NDFmls(v)$, and $FullLabel(v)$ are respectively called the available formulas of v , the negations of the formulas disallowed at v , and the full label of v . In an “and-or” graph, states play the role of “and”-nodes, while non-states play the role of “or”-nodes.

By the *local graph* of a state v we mean the subgraph of G consisting of all the path starting from v and not containing any other states. Similarly, by the local graph of a non-state v we mean the subgraph of G consisting of all the path starting from v and not containing any states.

Table 1. Some rules of the tableau calculus $C_{\mathcal{SHI}}$

$(\sqcap) \frac{X, C \sqcap D}{X, C, D}$	$(\sqcup) \frac{X, C \sqcup D}{X, C \mid X, D}$	$(H) \frac{X, \forall S.C}{X, \forall S.C, \forall R.C}$ if $R \sqsubseteq_{\mathcal{R}} S$
$(\exists) \frac{X, \exists R_1.C_1, \dots, \exists R_k.C_k}{C_1, X_1, \mathcal{T} \ \& \ \dots \ \& \ C_k, X_k, \mathcal{T}}$ if $\begin{cases} X \text{ contains no concepts of the} \\ \text{form } \exists R.D \text{ and, for } 1 \leq i \leq k, \\ X_i = \text{Trans}_{\mathcal{R}}(X, R_i) \end{cases}$		
$(\sqcap') \frac{X, a: (C \sqcap D)}{X, a:C, a:D}$	$(\sqcup') \frac{X, a: (C \sqcup D)}{X, a:C \mid X, a:D}$	$(H') \frac{X, a: \forall S.C}{X, a: \forall S.C, a: \forall R.C}$ if $R \sqsubseteq_{\mathcal{R}} S$
$(\forall') \frac{X, R(a, b)}{X, R(a, b), \text{Trans}(X, a, R, b), \text{Trans}(X, b, R^-, a)}$		
$(\exists') \frac{X, a_1: \exists R_1.C_1, \dots, a_k: \exists R_k.C_k}{C_1, X_1, \mathcal{T} \ \& \ \dots \ \& \ C_k, X_k, \mathcal{T}}$ if $\begin{cases} X \text{ contains no assertions of the} \\ \text{form } a: \exists R.D \text{ and, for } 1 \leq i \leq k, \\ X_i = \text{Trans}_{\mathcal{R}}(X, a_i, R_i) \end{cases}$		

We apply global state caching: if v_1 and v_2 are different states then $\text{Label}(v_1) \neq \text{Label}(v_2)$ or $\text{RFmls}(v_1) \neq \text{RFmls}(v_2)$ or $\text{DFmls}(v_1) \neq \text{DFmls}(v_2)$. If v is a non-state such that $\text{AfterTrans}(v)$ holds then we also apply global caching for the local graph of v : if w_1 and w_2 are different nodes of the local graph of v then $\text{Label}(w_1) \neq \text{Label}(w_2)$ or $\text{RFmls}(w_1) \neq \text{RFmls}(w_2)$ or $\text{DFmls}(w_1) \neq \text{DFmls}(w_2)$.

Our calculus $C_{\mathcal{SHI}}$ for the description logic \mathcal{SHI} will be specified, amongst others, by a finite set of tableau rules, which are used to expand nodes of tableaux. A *tableau rule* is specified with the following information: the kind of the rule (an “and”-rule or an “or”-rule); the conditions for applicability of the rule (if any); the priority of the rule; the number of successors of a node resulting from applying the rule to it, and the way to compute their contents.

Tableau rules are usually written downwards, with a set of formulas above the line as the *premise*, which represents the label of the node to which the rule is applied, and a number of sets of formulas below the line as the (*possible*) *conclusions*, which represent the labels of the successor nodes resulting from the application of the rule. Possible conclusions of an “or”-rule are separated by $|$, while conclusions of an “and”-rule are separated by $\&$. If a rule is a unary rule (i.e. a rule with only one possible conclusion) or an “and”-rule then its conclusions are “firm” and we ignore the word “possible”. The meaning of an “or”-rule is that if the premise is satisfiable w.r.t. \mathcal{R} and \mathcal{T} then some of the possible conclusions

are also satisfiable w.r.t. \mathcal{R} and \mathcal{T} , while the meaning of an “and”-rule is that if the premise is satisfiable w.r.t. \mathcal{R} and \mathcal{T} then all of the conclusions are also satisfiable w.r.t. \mathcal{R} and \mathcal{T} .

Such a representation gives only a part of the specification of the rules.

We write X, φ or φ, X to denote $X \cup \{\varphi\}$, and write X, Y to denote $X \cup Y$. Our *tableau calculus* C_{SHI} for SHI w.r.t. the RBox \mathcal{R} and the TBox \mathcal{T} consists of rules which are partially specified in Table 1 together with two special rules (*forming-state*) and (*conv*).

The rules (\exists) and (\exists') are the only “and”-rules and the only *transitional rules*. The other rules of C_{SHI} are “or”-rules, which are also called *static rules*. The transitional rules are used to expand states of tableaux, while the static rules are used to expand non-states of tableaux.

For any rule of C_{SHI} except (*forming-state*) and (*conv*), the distinguished formulas of the premise are called the *principal formulas* of the rule. The rules (*forming-state*) and (*conv*) have no principal formulas. As usually, we assume that, for each rule of C_{SHI} described in Table 1, the principal formulas are not members of the set X which appears in the premise of the rule.

Expanding a non-state v of a tableau by a static rule $\rho \in \{(\sqcap), (\sqcup), (\sqcap'), (\sqcup')\}$ which uses φ as the principal formula, we put φ into the set $RFmls(w)$ of each successor w of v . Recall that $RFmls(w)$ is called the set of the reduced formulas of w . If w is a non-state, $v_1 = ATPred(w)$ and $v_1, v_2, \dots, v_k = w$ is the path (of non-states) from v_1 to w , then an occurrence $\psi \in RFmls(w)$ means there exists $1 \leq i < k$ such that $\psi \in Label(v_i)$ and ψ has been reduced at v_i . After that reduction, ψ was put into $RFmls(v_{i+1})$ and propagated to $RFmls(v_k)$.

Expanding a simple (resp. complex) state v of a tableau by the transitional rule (\exists) (resp. (\exists')), each successor w_i of v is created due to a corresponding principal formula $\exists R_i.C_i$ (resp. $a_i : \exists R_i.C_i$) of the rule, and $RFmls(w)$ is set to the empty set.

For any state w , every predecessor v of w is always a non-state. Such a node v was expanded and connected to w by the static rule (*forming-state*). The nodes v and w correspond to the same element of the domain of the interpretation under construction. In other words, the rule (*forming-state*) “transforms” a non-state to a state. It guarantees that, if $BeforeFormingState(v)$ holds then v has exactly one successor, which is a state.

The rule (*conv*) used for dealing with converses will be discussed shortly.

The priorities of the rules of C_{SHI} are as follows (the bigger, the stronger): (\sqcap) , (\sqcap') , (H) , (H') , (\forall') : 5; (\sqcup) , (\sqcup') : 4; (*forming-state*): 3; (\exists) , (\exists') : 2; (*conv*): 1.

The conditions for applying a rule $\rho \neq (conv)$ to a node v are as follows:

- the rule has $Label(v)$ as the premise (thus, the rules (\sqcap) , (\sqcup) , (H) , (\exists) are applicable only to simple nodes, and the rules (\sqcap') , (\sqcup') , (H') , (\forall') , (\exists') are applicable only to complex nodes)
- all the conditions accompanying with ρ in Table 1 are satisfied
- if ρ is a transitional rule then $Type(v) = state$
- if ρ is a static rule then $Type(v) = non-state$ and

- if $\rho \in \{(\sqcap), (\sqcup), (\sqcap'), (\sqcup')\}$ then the principal formula of ρ does not belong to $RFmls(v)$, else if $\rho \in \{(H), (H'), (\forall')\}$ then the formula occurring in the conclusion but not in the premise of ρ does not belong to $AFmls(v)$
- no static rule with a higher priority is applicable to v .

We now explain the ways of dealing with converses, i.e., with inverse roles.

Consider the case when $Type(v) = \text{state}$, $SType(v) = \text{simple}$, $\exists R.C \in Label(v)$ and v corresponds to an element $x_v \in \Delta^{\mathcal{I}}$ of the interpretation \mathcal{I} under construction. We need to realize the formulas of $Label(v)$ at v so that $x_v \in (Label(v))^{\mathcal{I}}$. The formula $\exists R.C$ is realized at v by making a transition from v to w with $Label(w) = \{C\} \cup \text{Trans}_{\mathcal{R}}(Label(v), R) \cup \mathcal{T}$. The node w corresponds to an element $x_w \in \Delta^{\mathcal{I}}$ such that $(x_v, x_w) \in R^{\mathcal{I}}$ and $x_w \in C^{\mathcal{I}}$. If at some later stage we need to make $x_w \in (\forall R^-.D)^{\mathcal{I}}$ (for example, because $(\forall R^-.D) \in Label(w)$) then we need to make $x_v \in D^{\mathcal{I}}$, and hence we need to add D to $Label(v)$ as a requirement to be realized at v if $D \notin AFmls(v)$. Similarly, if at some later stage we need to make $x_w \in (\forall S.D)^{\mathcal{I}}$, where $R^- \sqsubseteq_{\mathcal{R}} S$ and $S \circ S \sqsubseteq S \in \text{ext}(\mathcal{R})$, then we need to make $x_v \in (\forall S.D)^{\mathcal{I}}$, and hence we need to add $\forall S.D$ to $Label(v)$ as a requirement to be realized at v if $\forall S.D \notin AFmls(v)$.

- If $x_v \in D^{\mathcal{I}}$ (where D may be of the form $\forall S.D'$) is a requirement but $D \notin AFmls(v)$ then we record this by setting $ConvMethod(v) := 0$ and add D to the set $FmlsRC(v)$. If $FmlsRC(v) \cap DFmls(v) \neq \emptyset$ then the requirements at v are unrealizable and we set $Status(v) := \text{unsat}$ (which means $\text{FullLabel}(v)$ is unsatisfiable w.r.t. \mathcal{R} and \mathcal{T}). If $FmlsRC(v) \neq \emptyset$ and $FmlsRC(v) \cap DFmls(v) = \emptyset$ then we set $Status(v) := \text{incomplete}$, which means the set $Label(v)$ should be extended with $FmlsRC(v)$ if v will be used.
- Consider the case when the computed set $FmlsRC(v)$ is empty. In this case, we set $ConvMethod(v) := 1$. Each node w_i in the local graph of w is an “or”-descendant of w and corresponds to the same $x_w \in \Delta^{\mathcal{I}}$ (for example, if $C_1 \sqcup C_2 \in Label(w)$ then we may make w an “or”-node with two successors w_1 and w_2 such that $C_1 \in Label(w_1)$ and $C_2 \in Label(w_2)$).
 - Consider the case $(\forall R^-.D) \in Label(w_i)$. Thus, $x_w \in (\forall R^-.D)^{\mathcal{I}}$ is one of possibly many alternative requirements (because w_i is one of possibly many “or”-descendants of w). If w_i should be selected for representing w and $D \notin AFmls(v)$ then we should add D to $Label(v)$. If $D \in DFmls(v)$ then we set $Status(w_i) := \text{unsat}$, which means the “combination” of v and w_i is unsatisfiable w.r.t. \mathcal{R} and \mathcal{T} .
 - Consider the case when $(\forall S.D) \in Label(w_i)$, $R^- \sqsubseteq_{\mathcal{R}} S$ and $S \circ S \sqsubseteq S \in \text{ext}(\mathcal{R})$. Thus, $x_w \in (\forall S.D)^{\mathcal{I}}$ is one of possibly many alternative requirements (because w_i is one of possibly many “or”-descendants of w). If w_i should be selected for representing w and $\forall S.D \notin AFmls(v)$ then we should add $\forall S.D$ to $Label(v)$. If $\forall S.D \in DFmls(v)$ then we set $Status(w_i) := \text{unsat}$, which means the “combination” of v and w_i is unsatisfiable w.r.t. \mathcal{R} and \mathcal{T} .

If, for $X = \text{Trans}_{\mathcal{R}}(Label(w_i), R^-) \setminus AFmls(v)$, we have that $X \neq \emptyset$ and $X \cap DFmls(v) = \emptyset$, then we add X (as an element) to the set $AltFmlSetsSCP(w)$ and set $Status(w_i) := \text{incomplete}$, which means that, if the “or”-descendant

w_i should be selected for representing w then X should be added (as a set) to $Label(v)$.

The case when $Type(v) = \text{state}$, $SType(v) = \text{complex}$ and $a: \exists R.C \in Label(v)$ can be dealt with in a similar way. See [10] for details.

When a node w gets status *incomplete*, *unsat* or *sat*, the status of every predecessor v of w will be updated as shown in procedure `UpdateStatus(v)` defined in [10, page 10]. In particular:

- If $Type(w) = \text{state}$ and $Status(w) = \text{incomplete}$ then `BeforeFormingState(v)` holds and w is the only successor of v . In this case, the edge (v, w) will be deleted and the node v will be re-expanded by the converse rule (*conv*) as shown in procedure `ApplyConvRule` given in [10, page 9]. For the subcase $ConvMethod(w) = 0$, we connect v to a node with label $Label(v) \cup FmlsRC(w)$. Consider the subcase when $ConvMethod(w) = 1$. Let $AltFmlSetsSC(w) = \{\{\varphi_1\}, \dots, \{\varphi_n\}, Z_1, \dots, Z_m\}$, where Z_1, \dots, Z_m are non-singleton sets. We connect v to successors w_1, \dots, w_{n+m} such that: for $1 \leq i \leq n$, $Label(w_i) = Label(v) \cup \{\varphi_i\}$, and for $n+1 \leq i \leq n+m$, $Label(w_i) = Label(v) \cup Z_i$. To restrict the search space, for $1 \leq i \leq n$, we add all φ_j with $1 \leq j < i$ to $DFmls(w_i)$. This can be read as: at v either allow to have φ_1 (by adding it to the attribute *Label*), or disallow φ_1 (by adding it to the attribute *DFmls*) and allow φ_2 , or disallow φ_1, φ_2 and allow φ_3 , and so on. Similarly, for $n+1 \leq i \leq n+m$, we add all φ_j with $1 \leq j \leq n$ to $DFmls(w_i)$.
- If $Type(v) = \text{state}$ (i.e. $Kind(v) = \text{and-node}$) and v has a successor w such that $Status(w) = \text{incomplete}$ then we set $AltFmlSetsSC(v) := AltFmlSetsSCP(w)$ and set $Status(v) := \text{incomplete}$.

Application of a tableau rule ρ to a node v is specified by procedure `Apply(ρ, v)` given in [10, page 8]. This procedure uses procedures `ApplyConvRule` and `ApplyTransRule` given in [10, page 9]. Procedures used for updating and propagating statuses of nodes are defined in [10, page 10]. The main function `Tableau($\mathcal{R}, \mathcal{T}, \mathcal{A}$)` is also defined in [10, page 10]. It returns a rooted “and-or” graph called a C_{SHI} -tableau for the knowledge base $(\mathcal{R}, \mathcal{T}, \mathcal{A})$. The root of the graph is a complex node ν with $Label(\nu) = \mathcal{A} \cup \{(a: C) \mid C \in \mathcal{T} \text{ and } a \text{ is an individual occurring in } \mathcal{A}\}$.

See the long version [10] of this paper for examples of “and-or” graphs and a proof of the following theorem.

Theorem 3.1. *Let $(\mathcal{R}, \mathcal{T}, \mathcal{A})$ be a knowledge base in NNF of the logic SHI. Then procedure `Tableau($\mathcal{R}, \mathcal{T}, \mathcal{A}$)` given in [10] runs in exponential time (in the worst case) in the size of $(\mathcal{R}, \mathcal{T}, \mathcal{A})$ and returns a rooted “and-or” graph $G = (V, E, \nu)$ such that $(\mathcal{R}, \mathcal{T}, \mathcal{A})$ is satisfiable iff $Status(\nu) \neq \text{unsat}$. \triangleleft*

4 Conclusions

We have given the first cut-free EXPTIME (optimal) tableau decision procedure for checking satisfiability of a knowledge base in the description logic SHI. Our

decision procedure is novel: in contrast to [3,4,5], it deals also with ABoxes; in contrast to [3,14], it does not use cuts; in contrast to [13,12], it deals also with inverse roles; and in contrast to [9], it deals also with transitive roles and hierarchies of roles. The procedure can be implemented with various optimizations as in [8] to give an efficient complexity-optimal program for checking satisfiability of a knowledge base in the popular DL \mathcal{SHI} .

Acknowledgements. This work was supported by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the strategic scientific research and experimental development program: “Interdisciplinary System for Interactive Scientific and Scientific-Technical Information”.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): Description Logic Handbook. Cambridge University Press, Cambridge (2002)
2. D’Agostino, M., Gabbay, D.M., Hähnle, R., Posegga, J. (eds.): Handbook of Tableau Methods. Springer, Heidelberg (1999)
3. Goré, R.P., Nguyen, L.A.: EXPTIME tableaux with global caching for description logics with transitive roles, inverse roles and role hierarchies. In: Olivetti, N. (ed.) TABLEAUX 2007. LNCS(LNAI), vol. 4548, pp. 133–148. Springer, Heidelberg (2007)
4. Goré, R., Widmann, F.: Sound global state caching for ALC with inverse roles. In: Giese, M., Waaler, A. (eds.) TABLEAUX 2009. LNCS, vol. 5607, pp. 205–219. Springer, Heidelberg (2009)
5. Goré, R., Widmann, F.: Optimal and cut-free tableaux for propositional dynamic logic with converse. In: Giesl, J., Hähnle, R. (eds.) IJCAR 2010. LNCS, vol. 6173, pp. 225–239. Springer, Heidelberg (2010)
6. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible \mathcal{SROIQ} . In: Doherty, P., Mylopoulos, J., Welty, C.A. (eds.) Proceedings of KR 2006, pp. 57–67. AAAI Press, Menlo Park (2006)
7. Horrocks, I., Sattler, U.: A description logic with transitive and inverse roles and role hierarchies. J. Log. Comput. 9(3), 385–410 (1999)
8. Nguyen, L.A.: An efficient tableau prover using global caching for the description logic \mathcal{ALC} . Fundamenta Informaticae 93(1-3), 273–288 (2009)
9. Nguyen, L.A.: Cut-free EXPTIME tableaux for checking satisfiability of a knowledge base in the description logic \mathcal{ALCI} . In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS(LNAI), vol. 6804, pp. 465–475. Springer, Heidelberg (2011)
10. Nguyen, L.A.: The long version of the current paper (2011), <http://arxiv.org/abs/1106.2305>
11. Nguyen, L.A., Szalas, A.: ExpTime tableaux for checking satisfiability of a knowledge base in the description logic \mathcal{ALC} . In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS(LNAI), vol. 5796, pp. 437–448. Springer, Heidelberg (2009)
12. Nguyen, L.A., Szalas, A.: Checking consistency of an ABox w.r.t. global assumptions in PDL. Fundamenta Informaticae 102(1), 97–113 (2010)
13. Nguyen, L.A., Szalas, A.: Tableaux with global caching for checking satisfiability of a knowledge base in the description logic \mathcal{SH} . T. Computational Collective Intelligence 1, 21–38 (2010)
14. Nguyen, L.A., Szalas, A.: ExpTime tableau decision procedures for regular grammar logics with converse. Studia Logica 98(3) (2011)

IT Business Standards as an Ontology Domain

Adam Czarnecki and Cezary Orłowski

Gdańsk University of Technology, Faculty of Management and Economics,
Department of Information Technology Management,
ul.Narutowicza 11/12, 80-233 Gdańsk, Poland
{Adam.Czarnecki,Cezary.Orlowski}@zie.pg.gda.pl

Abstract. The aim of this paper is to report a selection of Semantic Web aspects pertaining to ontology development activities in the domain of the IT business standard (TOGAF 9) such as formulating competency questions, conceptualization of the domain, resolution of the source knowledge deficiency and applying common design patterns in the OWL formalization. Authors also try to determine target groups that may benefit from such ontology models.

Keywords: enterprise architecture, information technology, ontology engineering, OWL, Semantic Web, standards, TOGAF.

1 Introduction

The old engineering joke says that the best thing about standards is that there are so many to choose from. But when a standard is chosen to be deployed, this deployment should conform to the guidelines provided in order to maintain compatibility. If not, we may have multiple implementations of the same norm that cannot be aligned.

Strict technical norms generally leave a narrow or no margin of interpretation and their scope, requirements, processes and outputs are well defined. But there is a range of IT business standards that operate on both sides of the borderline that separates technical and social systems. Their deployment usually relies on the context and often requires some tailoring before they can be applied in the organization. Also the standard description provided in documents may not be as precise as in the pure technical specification. Both aforementioned conditions may lead to the inconsistency of the solution that is based on that standard. The third issue one should bear in mind is that although standards are reviewed in order to eliminate all errors and discrepancies, there is still a margin of faults that may have been overlooked in the current revision.

There are numerous methods of enforcing the textual description of the standard with more formal structured information such as tables, figures and diagrams to mention a few. Authors of this paper would like to discuss the potential applicability of ontologies developed in the Web Ontology Language as one of the methods for assuring the consistency of standards. This approach would be considered from two perspectives: the standard originator and organization that is implementing the given standard.

Authors have chosen The Open Group Architecture Framework version 9 (TOGAF) as an example of the standard that combines business (social) and technical threads. To narrow the scope of the ontology domain, core content metamodel—a part of TOGAF standard that defines a formal structure of terms within enterprise architecture development method—has been selected. The paper focuses on the part of the ontology engineering process (however does not address any methodology in particular), i.e. on the set of activities: knowledge acquisition, conceptualization and formalization. This process serves as a case study for discussing the degree of the effort that must be put into reflecting the knowledge on the standard in the ontology and what benefits—if any—can be drawn from this project.

The intended readers of this paper are people involved in two kinds of processes related to standards: their devising and deploying. Authors would also like to notice that some earlier remarks on ontologies as tools for the IT management standards support were outlined in [2].

The first part of this paper briefly outlines TOGAF and its content metamodel. Then the activities in ontology engineering are described: competency questions formulation, domain knowledge acquisition based on TOGAF core content metamodel specification and the actual development of the ontology in OWL (which is tool-independent but in fact Protégé 4.0.2 application was used). To sum up, at the end of the paper conclusions are drawn and future works are outlined.

Due to the limited length of this paper a substantial part of the work (two figures and two tables) were excluded from the main text and published online as external references: [3], [7], [8] and [9]. Authors strongly suggest looking up to those links.

2 What is TOGAF

The Open Group Architecture Framework document specifies a detailed method and a set of supporting tools for developing enterprise architecture. The enterprise architecture (EA) can be described (attributive definition) as the organizing logic for key business process and IT capabilities reflecting the integration and standardization requirements of the firm's operating model [10]. The objective definition denotes a formal description of the phenomena given in the attributive definition. There is also a functional approach to the EA which indicates tasks and skills essential for managing the subject of the attributive definition [4].

TOGAF in its 9th version (later in this paper referred to as TOGAF 9 or TOGAF) addresses four main architectural domains:

1. Business,
2. Data,
3. Application,
4. Technology,

and there are six main parts of the framework:

1. Architecture Development Method (ADM)—a description of phases of the iterative cycle.
2. ADM guidelines and techniques—applying iteration to the ADM, its usage at the different enterprise levels.

3. Architecture content framework—specifies the content metamodel, artifacts, deliverables and building blocks.
4. Enterprise Continuum and tools—a specific repository of architectures and solutions.
5. Reference models—foundation architecture and integrated information infrastructure.
6. Architecture capability framework—reference material on how to manage organization structures, processes, roles, responsibilities and skills.

The idea of creating an ontology based on the TOGAF is not novel. SOA Working Group has created TOGAF 8 Ontology Draft [6] that captured most of the enterprise architecture artifacts and other objects and relations between them. It is important to add that TOGAF 8 was not equipped with a content metamodel description.

3 TOGAF Content Metamodel Overview

The TOGAF document is quite voluminous (700+ pages not including other referenced papers). TOGAF's Chapter 3 introduces 90 essential definitions as a prerequisite to the main part of the framework and there are 93 supplemental terms given in the Appendix A.

To define a formal structure for these terms and to ensure their consistency within the Architecture Development Method (ADM) content metamodel has been introduced in the TOGAF 9 document. The content metamodel defines a set of entities that allow architectural concepts to be captured, stored, filtered, queried, and represented in a way that supports consistency, completeness, and traceability [5, p. 373].

The TOGAF content metamodel is divided into the core metamodel which provides a minimum set of architectural content to support traceability across artifacts and metamodel extensions that provide concepts to support more specific or more in-depth modeling. In this paper the focus is set on the core content metamodel as a domain of the ontology.

4 Ontology Development

In following subsections the process of ontology development in the TOGAF core content metamodel domain has been described. For the sake of clarity of the example and because of the research being still in progress, the scope of the domain addresses the following parts of the content metamodel specification:

- Core content metamodel entities,
- Relationships between core content metamodel entities,
- Content metamodel objects (a list and a hierarchy but without relationships).

Among elements that have been omitted in this paper there are:

- ADM phases,
- Core architecture artifacts,

- Content metamodel extensions,
- Content metamodel attributes.

In the progress of further research the scope of the ontology should be broadened to encompass all concepts in TOGAF content metamodel.

The OWL DL formalization of the domain was implemented using Protégé 4.0.2 application. Authors have provided some examples of the OWL syntax in the text to better illustrate certain ontology constructs.

4.1 Competency Questions

The aim, scope and depth of the ontology depend on the knowledge it is supposed to store and conclude. The software engineering discipline has a requirements modeling technique of ‘use cases’ that a system should respond to. When specifying ontology a knowledge worker has a tool of ‘competency questions’—a list of issues that ontology should (must) reply to.

For the domain of TOGAF core content metamodel the following competency questions were formulated:

- What are the core metamodel entities?
- What is the structure of the core metamodel entities?
- How core metamodel entities relate to each other?
- What is the type of the individual that is described using object property assertion?

These questions were chosen to illustrate the process of ontology development and can easily be expanded by adding other elements of the core content metamodel and its extensions to the domain.

4.2 Core Content Metamodel Concepts

The TOGAF core content metamodel is the backbone (see: Fig. 34-1 in [5, p. 368]) of the content metamodel. The list of its concepts can be found on multiple pages of TOGAF9’s Chapter 34, e.g.:

- A list of core terms titled ‘Core Metamodel Entities’ on page 369,
- A Figure 34-2 titled ‘Core Entities and their Relationships’ on page 371,
- A Figure 34-5 titled ‘Detailed Representation of the Content Metamodel’ on page 375,
- A Figure 34-6 featuring a UML-like class diagram titled ‘Entities and Relationships Present within the Core Content Metamodel’ on page 376 (see: Fig. 1),
- A Figure 34-7 titled ‘Content Metamodel with Extensions’ on page 378,
- A Figure 34-8 titled ‘Relationships between Entities in the Full Metamodel’ on page 379,
- A table that lists content metamodel objects on pp. 393–396,
- A table with content metamodel attributes on pp. 396–406,
- A table of metamodel relationships on pp. 406–409.

This abundance of sources can be used for cross-checking of the list of core metamodel concepts that are to be stored as classes in the OWL ontology. This, however, reveals some ambiguity between sets of terms that need to be somehow resolved in the ontology (see: [8]). There are only four core metamodel concepts that appear through the mentioned sources and keep their names unchanged. These are: ‘Actor’, ‘Data Entity’, ‘Function’ and ‘Role’. Other concepts need preprocessing activities that are described below.

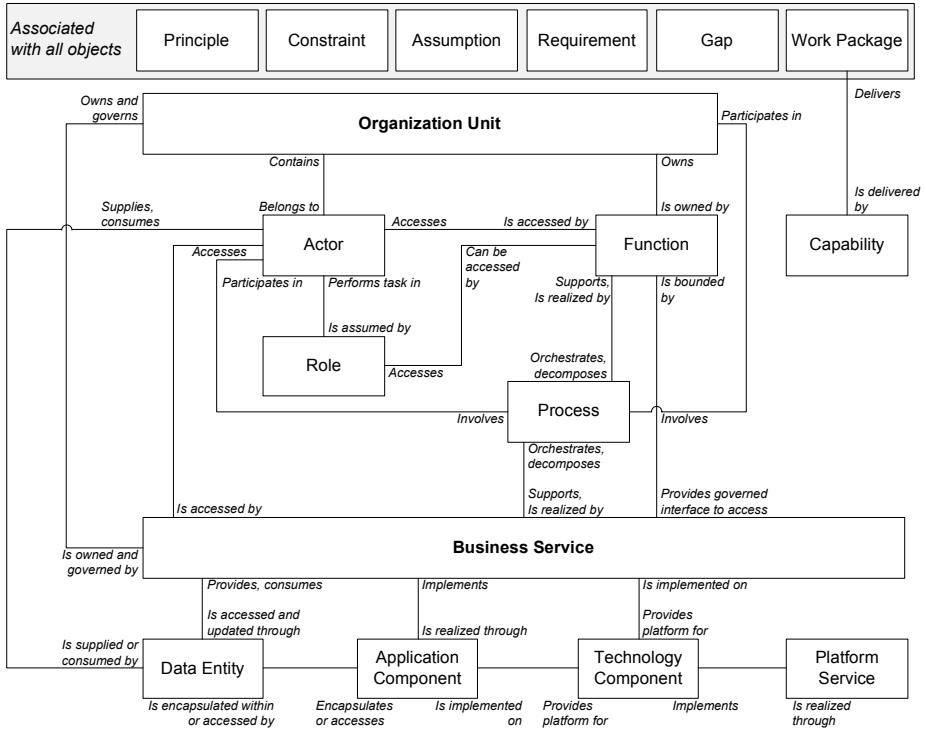


Fig. 1. Entities and Relationships present within the Core Content Metamodel [5, p. 376]

Concept Categories and Multiple Inheritance. One of the problems the TOGAF 9 ontology modeler meets is associated with the entity called an ‘Application Component’. It can be found on the core metamodel concepts list but later in the document this object splits into two: ‘Logical Application Component’ and ‘Physical Application Component’. The first one is marked as a part of the core content, the latter—a part of the infrastructure consolidation extension. All three entities are listed as metamodel objects, which includes core and extensions entities. The proposed solution to this problem would be to make ‘Logical Application Component’ and ‘Physical Application Component’ be both subclasses of the ‘Application Component’. The ‘Application Component’ would be a subclass of the ‘Content Metamodel Object’ class and ‘Logical Application Component’ would get a ‘Core Metamodel Entity’ as its superclass. The OWL syntax (in XML serialization) of these relations is presented below:

```
<owl:Class rdf:about="#Application_Component">
  <rdfs:subClassOf
rdf:resource="#Content_Metamodel_Object"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Logical_Application_Component">
  <rdfs:subClassOf
rdf:resource="#Application_Component"/>
  <rdfs:subClassOf
rdf:resource="#Core_Metamodel_Entity"/>
</owl:Class>
<owl:Class rdf:about="#Physical_Application_Component">
  <rdfs:subClassOf
rdf:resource="#Application_Component"/>
</owl:Class>
```

The OWL allows a class to have multiple superclasses, so there is no contradiction in the presented solution. A very similar situation to the given above applies to the set of ‘Technology Component’, ‘Logical Technology Component’ and ‘Physical Technology Component’ and has been resolved in the same manner.

Class Equivalence. The next problem encountered was the concept that had different names across the metamodel specification. In the presented example domain this applies to the pair of ‘Organization’ and ‘Organization Unit’. There are two constructs in the OWL language that can support the ontology engineer in stating that two classes represent the same concept: `owl:sameAs` and `owl:equivalentClass`. The first property links two individuals and can only be used in OWL Full to denote the identity of two classes. The second property, `owl:equivalentClass`, can be used for classes defined in the OWL DL and states that two classes are both of the same type (they are subclasses of each other which means that they share same sets of individuals).

Same pattern applies in the given domain to ‘Business Service’ and ‘Service’ classes. It may be misleading that there are three concepts present in the TOGAF core content metamodel: ‘Business Service’, ‘Platform Service’ and ‘Service’—the latter is not a superclass for Business and Platform Service but it is an abbreviated name of Business Service as one can conclude from cross-checking tables and diagrams in the TOGAF document.

Concepts Reduction. The example in this paper is limited to the domain of the core content metamodel in TOGAF 9 and therefore the list of [8] terms must be cleared. There are at least four forms of formality in the content metamodel description:

- Plain text,
- Lists (mostly bulleted, not numbered),
- Tables,
- Diagrams.

They have been listed here in the order of the ascending formality level. In general, the more formal the description is, the less interpretation is needed. So, when selecting candidates for classes in the ontology, diagrams (with UML classes and relations) and tables were mostly used. The final set of core metamodel entities is depicted on [7] (subclasses of the ‘Core Metamodel Entity’ class).

4.3 Content Metamodel Objects

All classes that are members of the ‘Core Metamodel Entity’ class are also members of the broader ‘Content Metamodel Object’ class, so although the ‘Core Metamodel Entity’ is not mentioned to be a formal content metamodel object, it has been declared to be the subclass of the ‘Content Metamodel Object’ class as shown on the [7] figure. It allows automatic classification of any future core metamodel entities to be also content metamodel objects. If it would be unwanted to have ‘Core Metamodel Entity’ class as a subclass of the ‘Content Metamodel Object’ one would have to assert that every subclass of the ‘Core Metamodel Entity’ is also a subclass of the ‘Content Metamodel Object’.

4.4 Metamodel Relationships

The section 34.7 of the TOGAF 9 specification contains a table of metamodel relationships with following columns:

- Source Object,
- Target Object,
- Name,
- Extension Module.

First two columns list objects that have been modeled in the ontology as subclasses of the ‘Content Metamodel Class’. The ‘Name’ column lists (mostly) relationships that were depicted on the fig. 34-8 in [5, p.379]. It corresponds well with the ‘subject–predicate–object’ triple that can be modeled in the OWL as a ‘class–object property–class’. But before names of the relationships will become object properties in the ontology, the above mentioned table should be preprocessed in order to:

- Select only relationships between core metamodel entities (as it narrows the scope to the given in this paper),
- Cross-check the consistency of relationships provided by a mentioned table and diagrams,
- Discover pairs of relationships that can be modeled as inversed object properties,
- Discover other types of object properties, e.g. functional, symmetric or transitive.

The output of the preprocessing activities is presented in [9]. If there was an inverse relationship between objects, the inverse relationship is not repeated in the second row with a switched source and target object.

The cross-checking of sources of knowledge of the core metamodel relationships has revealed the inconsistency between them. For instance, metamodel UML-like

diagrams (see: Fig. 1) mark the existence of associations between ‘Organization Unit’ and ‘Process’ entities while the section 34.7 of the TOGAF document—which lists metamodel relationships (see: [5, pp. 406–409])—lacks this information. None of the relationships pointing from the class to itself (such as ‘Decomposes’) were presented on any diagram. There are also distinct differences between core and full metamodel diagrams—in this case the full metamodel diagram relationships were accepted as they were coherent with the aforementioned table.

Relationships Categorization. There are 13 pairs of direct and inverse relationships with some appearing more than once in the metamodel. This includes 5 identified opportunities to create inverse relationships that were not present in TOGAF but may contribute to content metamodel. Most of these pair could be easily derived from the simple lookup: if ‘Object1–Relationship1–Object2’ and ‘Object2–Relationship2–Object1’, then Relationship1 and Relationship2 are inverse to each other. This, however, fails when one considers ‘Service’ and ‘Data Entity’ objects (see: [9]) when neither of two relationships directing from ‘Service’ to ‘Data Entity’ corresponds with the reversed relationship.

The OWL also supports symmetric object properties, where one relationship is direct and inverse at the same time. The candidate for the symmetric property is the ‘Communicates with’ relationship.

Modeling Relationships in the OWL. The first-class objects in the OWL that are suitable for storing relationships are object properties. The question arises how to model the triple of Object1–Relationship–Object2.

One of the approaches to the relationship modeling would suggest creating a triple of two individuals and one object property between them, i.e.:

```
<owl:ObjectPropertyrdf:about="#Delivers"/>
<owl:Thingrdf:about="#Capability"/>
<owl:Thingrdf:about="#Work_Package">
<Delivers rdf:resource="#Capability"/>
</owl:Thing>
```

As it is shown in the aforementioned code ‘Capability’ and ‘Work Package’ concepts are defined as individuals, not classes, due to the OWL DL limitations. To preserve concepts as classes an ontology engineer can use a number of design patterns. One of them is to set the domain and the range of the object property. This approach can be applied when the object property is and will be exclusively used to link entities of the given domain and range. If we had in the same ontology a domain–object property–range declaration that would state: ‘Programmer’–‘Delivers’–‘Source Code’, the reasoner would infer that a ‘Programmer’ is-a ‘Work Package’ and the ‘Source Code’ is-a ‘Capability’. And in fact, there is a number of relationships in the considered TOGAF domain that have the same name but links different objects, i.e. ‘Communicates with’, ‘Consumes’, ‘Decomposes’, ‘Is performed by’, ‘Is realized by’, ‘Orchestrates’, ‘Supplies’ and ‘Supports’. To resolve this name collision each of the mentioned relationships should become a superproperty of object properties that

link specific classes. At the time of writing this paper this task has not yet been completed, but the example for the Actor–Decomposes–Actor triple can be demonstrated is action: the definition of the ‘Actor decomposes Actor’ object property allows the reasoner to classify any individual that has an assertion that includes this property as a type of ‘Actor’ as shown on the referenced figure [3].

One could also use other OWL modeling technique: class equivalence of the existential restriction as a necessary and sufficient condition (definition) of the class. More on this subject can be found in [1].

4.5 Before the Next Iteration

During the process described in section 4 of this paper, an ontology consisting of 81 classes and 39 object properties has been developed. As it was mentioned earlier in the text, this ontology requires further work in developing and structuring more complex object properties and domain–range definitions.

The time required to build this ontology was 3–4 days including knowledge acquisition, conceptualization and formalization. The estimated time needed to finish the phase of core content relationship modeling is 0.5–1 day. Number of days needed to fully model TOGAF content metamodel is hard to assess but can range from 2 to 4 weeks for a single ontology engineer.

5 Summary and Future Works

The ontology development process described in this paper has shown that the domain of core content metamodel of TOGAF can be represented in the formal and computable language—OWL. This language allows for relatively easy knowledge classification based on existing reasoning software. It also allows detecting certain inconsistencies within the framework documentation. The next contribution of ontology is promoting sharing knowledge which collective computational intelligence can benefit from. The question arises whether the effort put in the ontology development brings benefits that could not be achieved in shorter time, with fewer resources engaged and with at least the same quality.

The results of applying ontology to a very narrow scope of the only one standard do not constitute a representative sample to draw general conclusions. However this experiment sheds some light on the path of the future research—the team involved in standard’s development may use ontology as a ‘back-office’ model to share a common vocabulary and understanding of the standard among the people involved in that collaboration. The parallel development of the ontology in that body could support the process of the standard elaboration and revision phases.

The second party that could benefit from the ontological model of the standard is organizations that intend to implement such standard. Although further studies are needed, the current experience of authors suggests that such effort of developing ontology in the domain of the given IT standard on the client-side is resource-consuming and requires skills that are rare on the job market. Therefore the economic efficiency of such activities would be low. But if the standard body would provide ontology of the standard—as it was suggested in the preceding paragraph—along with

the interface that would allow answering competency questions formulated by the client organization, the efficiency of the ontology use might increase. Authors of this paper would like to consider it as a 'working hypothesis' that needs further examination.

To conclude, IT business standards can be a promising array of domains of applying ontologies that may give these standards a sound formal semantic (logic) shape that is based on the Semantic Web architecture. It allows verification and validation of the model presented in the given standard. Yet, the degree in which such ontology can support the standards' bodies and users should be further studied.

References

1. Allemang, D., Hendler, J.: *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Morgan Kaufman, Burlington (2008)
2. Czarniecki, A., Orłowski, C.: *Ontology as a Tool for the IT Management Standards Support*. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) *KES-AMSTA 2010. LNCS(LNAD)*, vol. 6071, pp. 330–339. Springer, Heidelberg (2010)
3. *Reasoner Classification in Protégé Based on the Domain and Range Declaration (Appendix to this paper)*, <http://www.zie.pg.gda.pl/zbti/iccci2011/fig3.tif>
4. Sobczak, A.: *Modele i metamodely w architekturze korporacyjnej*. In: *Proceedings to The Organizations Support Systems Conference, Akademia Ekonomiczna w Katowicach, Katowice (2008)* (in Polish), http://www.swo.ae.katowice.pl/_pdf/396.pdf
5. *The Open Group Architecture Framework, Version 9*. The Open Group (2009)
6. *TOGAF 8 Ontology draft*. SOA Working Group, <http://www.opengroup.org/projects/soa-ontology/doc.tpl?gdid=11367>
7. *TOGAF Content Metamodel Objects (Appendix to this paper)*, <http://www.zie.pg.gda.pl/zbti/iccci2011/fig2.tif>
8. *TOGAF Core Metamodel Concepts According to Different Sources (Appendix to this paper)*, <http://www.zie.pg.gda.pl/zbti/iccci2011/tab1.htm>
9. *TOGAF Core Metamodel Relationships (Appendix to this paper)*, <http://www.zie.pg.gda.pl/zbti/iccci2011/tab2.htm>
10. Weill, P.: *Innovating with Information Systems: What do the most agile firms in the world do?* Presentation at Sixth e-Business Conference, Barcelona (March 27, 2007)

Attribute Selection-Based Recommendation Framework for Long-Tail User Group: An Empirical Study on MovieLens Dataset

Jason J. Jung and Xuan Hau Pham

Department of Computer Engineering
Yeungnam University
Dae-Dong, Gyeongsan, Korea 712-749
{j2jung, pxhauqbu}@gmail.com

Abstract. Most of recommendation systems have serious difficulties on providing relevant services to the “short-head” users who have shown intermixed preferential patterns. In this paper, we assume that such users (which are referred to as long-tail users) can play an important role of information sources for improving the performance of recommendation. Attribute reduction-based mining method has been proposed to efficiently select the long-tail user groups. More importantly, the long-tail user groups as domain experts are employed to provide more trustworthy information. To evaluate the proposed framework, we have integrated MovieLens dataset with IMDB, and empirically shown that the long-tail user groups are useful for the recommendation process.

Keywords: User modeling, Recommendation, Long-tail group, Attribute reduction.

1 Introduction

Efficient recommendation has been regarded as a key technology of various applications. In particular, all possible activities and feedbacks from online users should be collected, and efficiently analyzed to find out any meaningful relationships between users, between items, and between users and items. Thereby, most of such recommendation systems have been focusing on user modeling for comparing the users. A variety of user modeling methods [16][15][6] have been proposed for analyzing many types of user behaviors. In collaborative filtering methods for recommendation, user ratings are useful for modeling the corresponding users’ preferences. For example, as shown in Table 1, suppose that 4 users have rated 6 movies. By measuring similarity measures (e.g., Pearson correlation coefficient) between two arbitrary users, we can find that U_4 is most similar to U_1 . Hence, once U_1 has rated a good score (e.g., 5) to M_5 , the system can automatically recommend M_5 to U_4 .

Most of the recommendation systems are interested in only the group of users who have rated enough number of items (e.g., movies). We refer to the users as a *short head* group (i.e., U_1 , U_2 , and U_4). Consequently, it makes the systems possible to compare each set of ratings to the others. In contrast, recommending the rest of users like U_3

Table 1. Movie ratings by users

User ID	Movie ID (Title)	Rating
U ₁	M ₁ (Robin Hood)	5
	M ₂ (Inception)	2
	M ₃ (Avatar)	4
	M ₄ (Devil)	3
	M ₅ (Titanic)	5
U ₂	M ₃ (Avatar)	1
	M ₅ (Titanic)	2
U ₃	M ₂ (Inception)	5
	M ₆ (Memento)	3
U ₄	M ₁ (Robin Hood)	5
	M ₂ (Inception)	2
	M ₄ (Devil)	3

Table 2. Information sources about movie with three additional attributes

Movie ID (Title)	Genre	Actors	Directors
M ₁ (Robin Hood)	Action, Adventure, Drama	Russell Crowe, Cate Blanchett	Ridley Scott
M ₂ (Inception)	Action, Mystery, Sci-Fi, Thriller	Leonardo DiCaprio, Ellen Page	Christopher Nolan
M ₃ (Avatar)	Action, Adventure, Fantasy, Sci-Fi	Sam Worthington, Zoe Saldana	James Cameron
M ₄ (Devil)	Horror, Mystery, Thriller	Chris Messina, Logan Marshall-Green	John Erick Dowdle
M ₅ (Titanic)	Drama, History, Romance	Leonardo DiCaprio, Kate Winslet	James Cameron
M ₆ (Memento)	Crime, Drama, Mystery, Thriller	Guy Pearce, Carrie-Anne Moss	Christopher Nolan

is a serious challenge. These users, referred to as a *long tail* group, have rated only a few items and also peculiar items. Thus, it is even more difficult for the existing recommendation systems to find useful relationships between them. Due to the lack of user ratings from the long tail group, it is difficult to identify which item they are interested in. Also, more seriously, the chance to justify whether two users has common preferences is getting lower.

In fact, there have been some studies for extracting and also using the long tail. Most similarly, in Sathe et al. [13], a power user group has been selected for solving difficult problems in a specific medical domain. The users are assumed to have unique and outstanding knowledge and information. Meanwhile, a variety of methods, e.g., semantic query [7], fuzzy logic [4] and information integration [8], have been proposed. Especially, Brusilovsky et al. [1] has tried to merge several partial (and inexact) user models collected from multiple information systems.

In this paper, we focus on exploiting auxiliary information from external sources for identifying user preferences in the long tail group. For example, as shown in Table 2, if addition information about the movies is obtained, we can recognize that U₂ and U₃ have rated the only movies directed by ‘James Cameron’ (M₃ and M₅) and ‘Christopher Nolan’ (M₂ and M₆), respectively. Thus, once the system has new movie directed by the same directors, the movie will be recommended to appropriate users in the long tail group.

However, it is a problem to determine which attributes are working properly to discriminate the user preferences in the long tail. In order to deal with the problem, we propose a novel data integration framework to aggregate all available data sources for modeling the long tail groups more efficiently. Particularly, we focus on establishing several heuristics for measuring statistical patterns of the attributes. As a simple example in the previous Table 2 given two ratings by U_3 (i.e., M_2 and M_6), an attribute ‘Directors’ has shown the most dominant coverage (i.e., $\frac{|\text{Directors}(\text{Christopher Nolan})|}{2} = 1$), compared to any other attributes (e.g., $\frac{|\text{Genre}(\text{Action})|}{2} = 0.5$). The user can be recommended if there is an additional movies directed by “Christopher Nolan” as well as classified in “Action”.

In sum, we note that two main goals of this framework are

- to find which attribute is significant to identify long tail groups, and
- to exploit the experts of long tail groups for better recommendations to short head groups.

The outline of this paper is as follows. In the following Sect. 2 we explain the background knowledge about long tails, and show several existing work based on the long tail groups. Sect. 3 introduces several definitions for modeling users in long tail group, and present several heuristics for selecting significant attributes during data integration. Sect. 4 shows an experimental results for evaluating the proposed attribute selection methods and recommendation performance, and describes a case study on integrating MovieLens with IMDB datasets for identifying long tail groups. Finally, Sect. 5 draws our conclusions of this work.

2 Backgrounds and Related Work

In fact, the concept of long tail has found many different areas including online business, mass media, micro-finance, user-driven innovation, and social network mechanisms (e.g., crowdsourcing, crowdcasting, peer-to-peer), economic models, and marketing (viral marketing). The main assumption of this paper is that a group of users in long tail should be regarded as the professional experts in corresponding domains, and employed their opinions to provide recommendations to users in short head. In this section, we want to explain the background knowledge and previous studies about *i*) how to represent user interests and *ii*) how to recommend users.

2.1 Attribute-Based User Modeling

There have been a number of user modeling approaches, e.g., Bayesian network, neural networks and fuzzy logic [11]. Most recently, with emergence of semantic web communities, ontologies (or concept hierarchies, e.g., web directories) have been exploited to derive relevant ontological elements (e.g., concepts, properties, and instances) [5,6].

One of the simplest ways is to predefine a set of attributes, which are basis to measure the relevance to the user preferences. Also, the weight of each attribute can be computed to indicate the degree of the relevance to the user. If we assume that the attribute set A is a finite set, then this user model can be regarded as a vector form.

Definition 1 (User). Given a user rating $\mathcal{R}_i = \{\langle A, r \rangle | r \in [1, 5]\}$, a user u_i is simply represented as a set of attributes.

$$u_i = \{\langle a, w_a \rangle | a \in A_i, w_a = \frac{\sum r_a}{\text{occur}(a)}\} \quad (1)$$

where A_i is a finite set of attributes and w_a is a weight of an attribute a .

For example, in Table 1 the attributes can be a set of genres of movies, e.g., Action, Drama, and so on. The users can be represented as a set of genres into which the movies are classified. As a matter of fact, the more important issue is adaptability of user models by updating the weight of each attribute over time. Thus, this adaptive user modeling can capture the temporal changes of user interests, because the users have shown dynamic preference depending on many situations.

However, it is difficult to represent a number of attributes which are related with each other. As shown in Table 2, by nature, we can easily understand more meaningful relationships between the attributed.

2.2 Recommendation by User Classification

Once the users have been modeled, the recommendation system has to compare the user models for measuring similarities between two users. This is an essential assumption that most of systems are considering for providing recommendations. Thus, many social filtering methodologies and applications have been presented [3][12][14]. Especially, such approaches have been employed to distributed environments, e.g., adaptive learning community [2].

As shown in previous section, there are a number of user modeling methods. Of course, this comparison process is based on how the user model consists of. If we assume that all user models are represented as a set of attributes, then we can measure the similarity by using several heuristics [6].

Finally, these similarities between two users are applied to conduct user classification processes (e.g., k-means). A large number of users can be efficiently managed for propagating new recommendations.

3 Long-Tail User Group Selection

The attribute-based user modeling is efficient and easy to implicitly represent user preferences from user activities and feedbacks, only if scopes of the user preferences are identically limited. The attributes of a user should be selected from a predefined attribute set, so as to be matched with that of the other user.

However, if the users' feedbacks are not covered with the predefined attribute set, it is not possible for the user models to be compared with each other. To solve the problem, additional information can be exploited to capture the scopes of as many users as possible. Thus, there is even more chance of extract additional attributes from user behaviors.

For example, as shown in Table 2, the movie has many different attributes to be evaluated. When users are asked to rate the movies they watched, each of them might have different subjective criteria to determine whether the movies are good or not.

Thus, in this paper, we focus on Long Tail user Groups (LTuG) whose activities are more concentrated into a smaller set of attributes. Opposite to LTuG, Short Head user Groups (SHuG) have shown so diverse activities that the number of attributes are higher. In order to discriminate these two user groups and select the LTuG, we need to integrate external information (e.g., Table 2).

3.1 Dominant Attributes

In this paper, we want to figure out what is main motivation to take a certain action. In case of watching movies, ones have different opinions to choose movies. Two users U_1 and U_3 have watched the same movie M_2 in Table 1. We can find out that U_3 has chosen M_2 because the director is “Christopher Nolan.” If there is a new movie by the same director, the system has to recommend the movie to him. In this context, “Christopher Nolan” is a dominant attribute to represent his user model.

Hence, two methods can be presented in this paper to select the dominant attributes. First method is to measure a “dominant coverage” score of each attribute from the user ratings for efficiently justifying which attribute is strongly related to the user preference.

Definition 2 (Dominant coverage). *Given a user u_i , a dominant coverage τ of each attribute can be measured by*

$$\tau_i(A) = \mu_{a \in A} \left(\frac{\text{occur}(a, \mathcal{R}_i)}{|\mathcal{R}_i|} \right) \quad (2)$$

where \mathcal{R}_i is a set of ratings by the user, and μ is a function to compute a mean value.

More importantly, we do not want to consider how the users rate the items. Even though some of users have decided to rate movies with low scores, they may be interesting on some attributes related to the movies.

As shown in Table 3, the dominant coverage of each item, which is composed of an attribute, can be measured without taking into account how the users rate the items. Then, finally, we can compute the dominant coverage of the attribute by aggregating the information about the items. For example, attribute ‘Director’ of U_1 can be assigned with

$$\tau_i(Director) = \frac{0.2 + 0.2 + 0.4 + 0.2}{4} = 0.25 \quad (3)$$

where the ‘Director’ consists of four items. Roughly, we can understand that U_2 and U_3 have shown strong interests on ‘Director’ attribute (which are J. Cameron and C. Nolan, respectively), while U_4 does not have any dominant attribute explicitly.

Once we obtained the dominant coverage of attributes, we formulate two possible heuristics for extracting dominant attributes.

Table 3. An example of computing the dominant coverages of attributes

Users	Genre	$\tau(Genre)$	Actors	$\tau(Actors)$	Director	$\tau(Director)$
U ₁	Action	0.176	R. Crowe	0.1	R. Scott	0.2
	Adventure	0.118	C. Blanchett	0.1	C. Nolan	0.2
	Drama	0.059	L.DiCaprio	0.2	J.Cameron	0.4
	Mystery	0.118	E. Page	0.1	J.E. Dowdle	0.2
	Sci-Fi	0.118	S. Worthington	0.1		
	Thriller	0.118	Z. Saldana	0.1		
	Fantasy	0.059	C. Messina	0.1		
	Horror	0.059	L. Marshall-Green	0.1		
	History	0.059	K. Winslet	0.1		
U ₂	Action	0.143	L. DiCaprio	0.25	J.Cameron	1
	Adventure	0.143	S. Worthington	0.25		
	Drama	0.143	Z. Saldana	0.25		
	Sci-Fi	0.143	K. Winslet	0.25		
	Fantasy	0.143				
	History	0.143				
U ₃	Action	0.125	L. DiCaprio	0.25	C.Nolan	1
	Drama	0.125	E. Page	0.25		
	Mystery	0.25	G. Pearse	0.25		
	Sci-Fi	0.125	C.-A. Moss	0.25		
	Thriller	0.25				
	Crime	0.125				
U ₄	Action	0.2	R. Crowe	0.167	R. Scott	0.33
	Adventure	0.1	C. Blanchett	0.167	C. Nolan	0.33
	Drama	0.1	L. DiCaprio	0.167	J.E. Dowdle	0.33
	Mystery	0.2	E. Page	0.167		
	Sci-Fi	0.1	C. Messina	0.167		
	Thriller	0.2	L. Marshall-Green	0.167		
	Horror	0.1				

Definition 3 (Dominant attribute). A dominant attribute of a user U_i is selected when its dominant coverage is significantly larger than the others. Hence, a set of dominant attributes can be represented as the following two methods

$$A_i^T = \left\{ A_j \mid \max_{A_k \in \mathcal{A}} \tau(A_k) \right\} \quad (4)$$

$$= \{ A_j \mid \tau(A_k) \geq \mu_{A \in \mathcal{A}}(\tau(A_i)) \} \quad (5)$$

where \mathcal{A} is a set of all attributes.

Table 4. An example of selecting the dominant attributes from Table 3

Heuristics	Eq. 4	Eq. 5
U ₁	{ Director = "J. Cameron" }	{ Director = "J. Cameron" }
U ₂	{ Director = "J. Cameron" }	{ Actors = *, Director="J. Cameron" }
U ₃	{ Director = "C. Nolan" }	{ Director = "C. Nolan" }
U ₄	{ }	{ Director = * }

As shown in Table 4, the first heuristic (i.e., Eq. 4) allows to choose the only attribute whose dominant coverage is maximum. Unfortunately, this heuristic can not discover any dominant attributes for U₄, because the user's ratings are evenly distributed on all items (i.e., no significant patterns to be discovered). On the other hand, the second heuristic (i.e., Eq. 5) can employ a mean value as a threshold for filtering out. This heuristic can more sophisticatedly extract the dominant attributes that the first one.

Next, the second method of extracting dominant attributes is to investigate a temporal change by measuring variance during collecting the user ratings. We expect that a variance of dominant coverage for a certain duration will be playing an important role for detecting the dominant attributes. As a user keeps rating more items over time, his interests will be expressed more clearly. Thus, it is important to realize how the preferences of a user are stable over time. Especially, in an initial stage, it is difficult to precisely measure the dominant coverage (i.e., the first heuristic), due to the lack of user ratings at the moment. It is also called the cold start problem.

Thus, the coverage variance of each attribute ρ can be formalized.

Definition 4 (Coverage variance). *Given a set of ratings from a user U_i during $[t_0, t_T]$, the coverage variance can be given by*

$$\rho_i(A)_{[t_0, t_T]} = \text{var}_{t=t_0}^{t=t_T} (\tau_i^{(t)}(A)) \quad (6)$$

where $A \in A_i^\tau$ and var is a function to computer a variance value.

The coverage variance can be exploited to justify whether a certain dominant attribute is being converged or not. The given time duration is segmented into several intervals, and the coverage variance should be measured in each time interval. For example, suppose that the number of time intervals is 2. We only consider that the attribute A is a dominant attribute of user U_i , if $\rho_i(A)_{[t_0, t_1]} \leq \rho_i(A)_{[t_1, t_2]}$. Finally, the set of dominant attributes is denoted as $A^{\tau+\rho}$.

Once we have computes these two measurements (i.e., τ and ρ) of each attribute from the users, we can justify who the long tail user groups are.

Definition 5 (Long tail user group). *Given a number of users and their ratings, a long tail group LTuG is represented as*

$$\text{LTuG} = \{U_i | A_i^{\tau+\rho} \neq \phi\} \quad (7)$$

where ϕ indicates an empty set.

3.2 Recommending Short Head Users

We are regarding the LTuG as an expert group who is strongly interested in particular attributes (e.g., a director and an actor). In this paper, we want to exploit them to help to recommend the short head user group (SHuG).

The recommendation system needs to provide relevant information about a certain attribute to the short head user group. Thus, the basic idea of recommending the SHuG is to reuse the user rating given by the LTuG. To provide $U_j \in \text{SHuG}$ with the information about an attribute A_q , we conduct the following steps;

1. We can retrieve a set of users from LTuG, whose dominant attributes include A_q , by justifying the following condition

$$A_q \in A_j^{\tau+\rho} \quad (8)$$

where it returns boolean value.

2. The ratings of the SHuG can be integrated with the ratings from a set of users from LTuG, as follows;

$$\mathcal{R}_j \leftarrow \mathcal{R}_j + (\cup_i \mathcal{R}_i) \quad (9)$$

where it returns a whole rating set.

The integrated user ratings are simply assumed to be applied to normal recommendation schemes (e.g., collaborative filtering). We want to show that the performance of the normal recommendation scheme can be improved by the integration process to the SHuG.

4 A Case Study and Experimental Results

In this section, we want to describe how we have evaluated the proposed LTuG-based recommendation system. As a case study, movie recommendation has been chosen for applying the proposed method. To do so, we have collected MovieLens dataset¹ as user ratings, and investigated a set of attributes from Internet Movie Database².

We have selected 8 attributes from IMDb, as follows.

- Genre = { Action, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Family, Fantasy, Film-Noir, Game-Show, History, Horror, Music, Musical, Mystery, News, Reality-TV, Romance, Sci-Fi, Sport, Talk-Show, Thriller, War, Western }
- Director (i.e., Multiple directors are allowed.)
- Actors (i.e., Maximum 5 top actors have been selected from the list.)
- Storyline (i.e., Maximum 5 keywords have been selected by TF-IDF analysis.)
- Plot keywords
- Production company
- Country
- Language

It is important to confirm whether the user rating dataset can contain the long tail user groups in the real case. We have found out that *i*) attribute “director” is the most important dominant attribute, and *ii*) about 17.8% of users have been regarded as a long tail user group.

5 Conclusion and Future Work

As a conclusion, we have proposed a novel user modeling method for long tail users. The main contribution of this work is to efficiently establish the long tail users who can be regarded as expert group on a certain attribute. Hence, the user ratings and feedback given by this long tail user groups have been exploited to provide more relevant recommendation to non-expert user group, called short head group. Moreover, as additional

¹ <http://movielens.umn.edu/>

² IMDb, <http://www.imdb.com/>

contribution, we have shown the data integration scheme (e.g., MovieLens and IMDb) which can extract the meaningful but hidden attributes for user modeling.

In future work, dynamic patterns of user ratings should be more studied, because user ratings are a kind of streaming data over time. Also, there are a number of different patterns to be investigated [10]. Again, we want to collect more case studies to consider the long tail phenomena in the context of recommendation. Another important issue is to integrating information from heterogeneous sources. We are expecting to collect real ontologies from ontology-based recommendation systems [9], and integrate them for better user modeling.

Acknowledgments. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2011-0017156).

References

1. Brusilovsky, P., Sosnovsky, S., Yudelson, M., Kumar, A., Hsiao, S.: User model integration in a distributed adaptive e-learning system. In: Berkovsky, S., Carmagnola, F., Heckmann, D., Kuflik, T., Krüger, A. (eds.) Proceedings of the 6th International Workshop on Ubiquitous User Modeling, pp. 1–10 (2008)
2. del Olmo, F.H., Gaudioso, E., Boticario, J.: A multiagent approach to obtain open and flexible user models in adaptive learning communities. In: Brusilovsky, P., Corbett, A.T., de Rosi, F. (eds.) UM 2003. LNCS, vol. 2702, pp. 203–207. Springer, Heidelberg (2003)
3. Fisk, D.: An application of social filtering to movie recommendation. In: Nwana, H.S., Azarmi, N. (eds.) Software Agents and Soft Computing: Towards Enhancing Machine Intelligence Concepts and Applications. LNCS, vol. 1198, pp. 116–131. Springer, Heidelberg (1997)
4. John, R.I., Mooney, G.J.: Fuzzy user modeling for information retrieval on the world wide web. *Knowledge and Information Systems* 3(1), 81–95 (2001)
5. Jung, J.J.: Ontological framework based on contextual mediation for collaborative information retrieval. *Information Retrieval* 10(1), 85–109 (2007)
6. Jung, J.J.: Ontology-based context synchronization for ad-hoc social collaborations. *Knowledge-Based Systems* 21(7), 573–580 (2008)
7. Jung, J.J.: Query transformation based on semantic centrality in semantic social network. *Journal of Universal Computer Science* 14(7), 1031–1047 (2008)
8. Jung, J.J.: Consensus-based evaluation framework for cooperative information retrieval systems. *Knowledge and Information Systems* 18(2), 199–211 (2009)
9. Jung, J.J.: Ontology mapping composition for query transformation on distributed environments. *Expert Systems with Applications* 37(12), 8401–8405 (2010)
10. Jung, J.J.: Reusing ontology mappings for query segmentation and routing in semantic peer-to-peer environment. *Information Sciences* 180(17), 3248–3257 (2010)
11. Kok, A.J.: A review and synthesis of user modelling in intelligent systems. *Knowledge Engineering Review* 6(1), 21–47 (1991)
12. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International World Wide Web Conference (WWW 2010), pp. 285–295. ACM, New York (2001)
13. Sathe, N.A., Lee, P., Giuse, N.B.: A power information user (piu) model to promote information integration in tennessee’s public health community. *Journal of Medical Library Association* 92(4), 459–464 (2004)

14. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. *Advances in Artificial Intelligence* 2009(4), 2 (2009)
15. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Feature-Weighted User Model for Recommender Systems. In: Conati, C., McCoy, K., Paliouras, G. (eds.) *UM 2007. LNCS(LNAI)*, vol. 4511, pp. 97–106. Springer, Heidelberg (2007)
16. Zhang, Y., Koren, J.: Efficient bayesian hierarchical user modeling for recommendation system. In: Kraaij, W., de Vries, A.P., Clarke, C.L.A., Fuhr, N., Kando, N. (eds.) *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 47–54. ACM, New York (2007)

IOEM - Ontology Engineering Methodology for Large Systems

Joanna Sliwa¹, Kamil Gleba¹,
Wojciech Chmiel², Piotr Szwed², and Andrzej Glowacz²

¹ Military Communication Institute, Zegrze, Poland
{j.sliwa,k.gleba}@wil.waw.pl

² AGH University of Science and Technology, Krakow, Poland
{wch,pszwed,aglowacz}@agh.edu.pl

Abstract. The paper presents IOEM, a methodology for ontology development elaborated for the INSIGMA project. Although prepared for a particular use, the methodology is quite general and can be used in a large variety of IT projects requiring ontology components. It is particularly suitable for large and geographically distributed software projects. The methodology is oriented towards applications of ontologies in various phases of a software lifecycle: development and run-time.

Keywords: methodologies for building ontology, ontology engineering methodology, approaches for building ontologies for complex systems, ontology requirements, ontology life cycle.

1 Introduction

IOEM is an ontology engineering methodology designed for the Intelligent System for Global Monitoring Detection and Identification of Threats. The system is being developed within the INSIGMA project carried out by four Polish academic, research and commercial bodies (University of Science and Technology (AGH) - the consortium leader, Military Communication Institute (MCI), Military University of Technology (WAT) and University of Computer Engineering and Telecommunication (WSTKT)). The effect of the project will be a complex monitoring system used to identify objects in the monitored environment and, based on the stored information and advanced algorithms, identify threats related to both - the traffic and suspicious behaviour of people. The system can be used for traffic management and route planning for individual users and for the public safety services as well. The route planning will also take into account complex parameters that provide the possibility to select the route in special circumstances, e.g. after a road accident or a natural disaster, in difficult weather conditions, etc.

INSIGMA system will store and process large amounts of various types of data. Some of them will be raw data flowing from the sensors, some will have to be fused and processed in order to provide additional information. Moreover,

the identification of threats requires methods for automatic recognition and classification of events in the system. One of the main tasks within the INSIGMA project is thus to define ontology that would help to manage the information and, at the same time, would help in automatic identification and classification tasks.

The objective of the article is to present INSIGMA Ontology Engineering Methodology (IOEM) designed for the large system with geographically dispersed groups of developers.

2 Application of Ontologies in INSIGMA System

One of the key assumptions about the INSIGMA system is that it will be developed as an ontology driven information system. The role of ontologies for such systems can be classified in two dimensions: temporal and structural [1]. The temporal dimension is related to the stage in the software lifecycle: development time or run time. The structural dimension concerns the usage of ontologies in particular components: databases, user interface and business logic layer.

Ontologies developed within INSIGMA project fall into two categories: domain ontologies, specifying concepts of a particular domain (e.g. routes, vehicles, elements of a monitoring infrastructure, threats, weather) and task ontologies related to tasks, activities, processes (e.g. threats detection, services configuration, route planning). A particular software component can use both types of closely related domain and task ontologies.

Ontologies at the run time are used in situations where the domain model cannot be fully elaborated during the system development (some domain aspects are unknown or uncertain), or a kind of reasoning is required. In case of INSIGMA system, main uses are: classification of threats, their properties, reasoning about incidents, dynamic configuration of the system architecture to fit particular needs of an end-user, provision of semantic interoperability between components, that cannot be defined during the design stage and integration with external systems.

3 Ontology Engineering Process and Related Work

Methodologies for ontology engineering have been subject of research for a number of years. In general, ontology development depends on the context and the purpose of particular project. Therefore, each project team is very often trying to build its own methodology. The basis for the methodology development for INSIGMA system was the analysis of the most known existing approaches in that field.

There are many different ontology engineering methodology proposals. We analyzed typical methodologies used to build ontologies from scratch or by reusing other ontologies. In particular, the approaches dealt with were: On-ToKnowledge Methodology [2], Ontology development proposed by Noy and McGuinness [3],

Enterprise Ontology [4], TOVE (TOronto Virtual Enterprise) [5], HCONE (Human Centered ONtology) [6], UPON (Unified Process for ONtology building) [7], Holsapple [8], METHONTOLOGY [9].

After analyzing the above methodologies it was possible to distinguish some common activities (processes) for all methodologies which form an ontology life cycle:

- management process: consists of scheduling, control and quality assurance,
- ontology development process: consists of environment study, feasibility study specification, conceptualization, formalization, implementation, maintenance and use,
- support process: consists of activities run in parallel to the ontology development process. It includes knowledge acquisition, evaluation, integration, documentation, merging, configuration and alignment.

Each methodology defines its individual approach to carrying out the complete ontology life cycle. The above-mentioned groups of processes were incorporated in the proposed IOEM.

4 Requirements for INSIGMA Ontology

Generally, it was assumed that the ontology for INSIGMA system should be explicit, coherent and extensible.

Being explicit means that an ontology should efficiently communicate intended meaning of defined terms. Definitions of terms should be objective and insusceptible to unintended interpretations.

Being coherent means that an ontology should allow for drawing meaningful inferences that are consistent with definitions and axioms.

Being extensible means that an ontology developer should take into account future extensions of created ontology without the need for revising definitions. New terms in ontology should be able to be defined based on the existing vocabulary. It should also be open to other existing ontologies. In this case it is important to specify the upper ontology that would enable definition of additional domain ontologies not colliding with the existing ones.

5 INSIGMA Ontology Engineering Methodology

5.1 The Approach

The IOEM is focused on the ontology application process. It is not intended to develop upper ontologies or specific ontologies used to reflect knowledge in particular domain, but ontologies used to realize the goal of the project.

The ontology engineering process presented in this chapter is in many cases similar to the object-oriented analysis and requirements management. It is a natural consequence of the fact, that for a given set of applications UML (Unified Modeling Language) [10] specifications and semantic models have similar role on

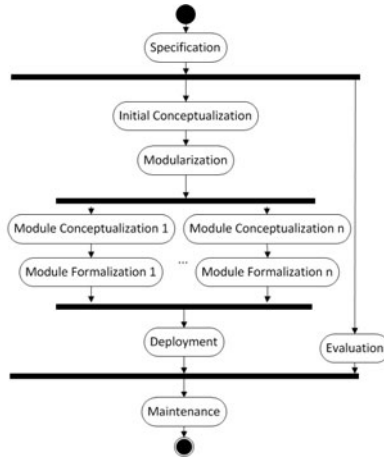


Fig. 1. IOEM stages

software engineering process. Following steps of IOEM have been presented in the subsequent subchapters: Specification, Initial Conceptualization, Modularization, Module Conceptualization, Formalization, Deployment, Evaluation and Maintenance. These basic steps have been presented in Fig. 1.

5.2 Specification Phase

The objective of the Specification phase is to determine the domain, scope of the ontology and aim of its application. The basis for this phase is the identification of the problem that the ontology application is to solve. This is very important in large scale systems since, at the beginning of the project, some of the developers do not know what is the final result they want to achieve and tend to support the idea of developing very broad and extensive ontology, which in turn may be inconvenient and rarely used. Therefore it is a good practice therefore to define a questionnaire for the developer teams consisting of a set of questions and exercises. This is to help to define a strong foundation for the ontology. An example questionnaire defined for INSIGMA consists of the following elements:

– **Ontology foundation**

- What is the purpose of creating an ontology ?
- What is the domain of the developed ontology?
- For what types of questions should the ontology provide answers?
- Who will be end-users of the ontology?
- Who will be responsible for the ontology maintenance?
- What are ontology use cases in the INSIGMA system?

– **Defining competency questions and motivating scenarios with use cases**

Questions that an ontology should be able to answer are called competency questions. After the ontology is built, competency questions enable it to be verified in terms of its completion.

Motivating scenario describes requirements that the created ontology should satisfy expressed as use cases. They provide examples for ontology application in the system.

– **Examples of existing ontologies application**

Large number of ontologies have been developed and published. Using them (importing) into the INSIGMA ontology may accelerate the work, reduce costs and support interoperability. The potential candidates should be evaluated to check their validity and possibility for refinement and extension for particular domains and tasks.

The elements of the questionnaire are very important in the process of ontology development. For instance, competency questions determine the ontology application, which is crucial, e.g. for rule-based systems and expert systems focused on solving particular problems. Two ontologies describing the same domain of knowledge may give answers to completely different questions. This task should be therefore considered very important since it strongly influences the resulting ontology and can considerably scope it further usage. Reusing existing ontologies is also beneficial and should be taken into account. This supports the interoperability, enables reusing of the results of other projects and can decrease the amount of work for ontology development.

5.3 Initial Conceptualization Phase

The goal of the Initial Conceptualization phase is to determine the scope of the developed ontology by establishing a glossary of key concepts, their hierarchy and relations. The result of this phase is usually an informal ontology description that may take: a narrative form, partial UML class diagrams or mind map diagrams. At this stage it is also possible to create a small monolithic ontology coded in a formal language, e.g. OWL (Web Ontology Language) [11], covering only main concepts and relations from the modeled domain or providing a template that can be applied while building the whole model.

In particular, the elements of the domain model do not have to be anchored in an upper ontology, many concepts can be omitted or not linked by relations. An important activity related to the Initial Conceptualization is the assessment whether the scope of the model is sufficient to answer the competency questions and support the use cases realization. This activity is considered to be a part of the Evaluation process running in parallel with other processes.

At the end of this phase, the team responsible for the ontology development should attain a consensus concerning the ontology scope (detailing what should be included in the ontology) and guidelines how it should be expressed. The first issue is subject of interest of domain experts, while the second one of knowledge engineers who are responsible for ontology formalization carried out in further stages.

5.4 Modularization Phase

Large ontologies are rarely constructed in a monolithic form, they are rather distributed into several modules (compound ontologies). Modularization is a natural method for management of complex models [12][13]. Depending on a starting point, it is achieved by decomposition (top-down approach) or composition (bottom-up approach). Decomposition is more often used while describing a behavior, e.g. functional decomposition depicted by DFD (Data Flow Diagrams), whereas composition is usually used to describe structural relations, e.g. building a class from compound classes (attributes), organizing classes into packages, modules or libraries and using them from outside the organization unit.

In case of ontologies expressed in OWL language, a composition of ontologies is supported by the import mechanism. It allows to include external ontologies published in the web or stored in a local library into a master ontology and gain access to their content. Imported ontologies may contain a taxonomy of classes, a taxonomy of relations, axioms, individuals, rules or any coherent combinations of these elements.

The result of import is an import closure. The import closure for an ontology O is a set containing all entities of O and entities of all ontologies that are directly or indirectly imported by O . It should be noted however, that if two ontologies $O1$ and $O2$ are imported into a master ontology, they should not contain different versions of the same concepts or relations (it is forbidden to import different versions of the same ontology) or explicit indication, that they are incompatible (with use of the annotation `owl:incompatibleWith` referring to the URI of another ontology).

The goal of Modularization is:

- reuse of existing components,
- logical distribution of the ontology building process into a set of partially independent activities,
- giving an opportunity to parallel development of compound modules; this issue is particularly important for a team being distributed among several geographical locations, which is the case of the INSIGMA project,
- providing a possibility of scaling (several useful closures can be created by importing selected components, they may have different applications),
- testing of developed ontologies (e.g. by creating an ontology for testing purposes that imports entities from a developed ontology and adds several individuals to validate rules or to reason about relations).

Fig. 2 shows a metamodel specifying entities relevant to the Modularization phase. An ontology may contain classes (concepts), properties (relations), etc. An ontology may import other ontologies and may have some `CompetencyQuestions` specified. The class `OntologyApplication` describes a particular application of ontology important from the perspective of the system development or deployment. The application is related to a selected ontology (more precisely an import closure). For a set of ontologies, there may exist several applications.

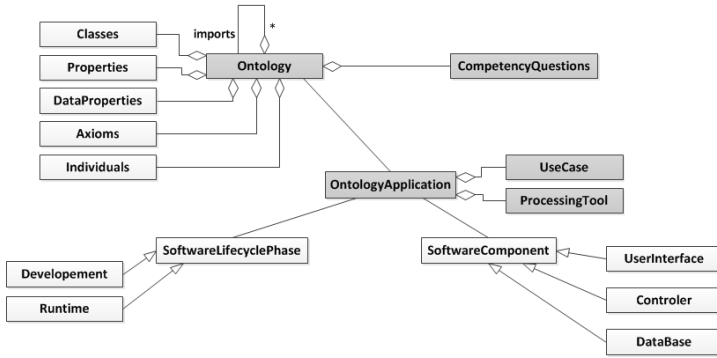


Fig. 2. INSIGMA Ontology and its applications

An ontology application:

- is related to a software lifecycle phase (Development or Runtime),
- concerns a particular software component (a user interface, a data base, a controller module),
- may require an additional tool for ontology processing or deployment.

Formally, in IOEM methodology the elements of the metamodel are represented by an artifact: the Ontology Application Chart. The document taking a form of questionnaire is filled out by two independent teams: the one developing an ontology and the second deploying it. Ontology Application Chart contains the following elements:

- architecture specification: indication of the master and compound ontologies,
- specification of the ontology role in reference to the software lifecycle and the software architecture,
- refinement of use cases (it should be mentioned that for ontologies used during the run-time, their use cases correspond to the system use cases, whereas for ontologies used at the development phase, the use cases are related to supporting tools for building and processing ontologies),
- the status of the supporting tools (if applicable),
- methods of ontology storing,
- project roles responsible for ontology development, deployment and maintenance.

5.5 Module Conceptualization Phase

Conceptualization is an activity that should be carried out before the Formalization process. It provides a domain ontology model in an informal language with terms and their relationships presented in ordered taxonomy. This process consists of the following actions: building a glossary of terms, building taxonomies of concepts, defining relations (between concepts and data types), building concept dictionary, defining axioms, defining rules, creating instances of particular classes.

To support the Conceptualization phase the authors proposed the second questionnaire, which helps to build conceptual dictionary and to define rules that describe the given domain.

5.6 Formalization Phase

This phase transforms the conceptual model into the formal model, which enables to automatically process the knowledge and verify its consistency. The phase includes the building of a semantic model in an ontology language. It is important to choose an appropriate formal model for the purpose of the project. It needs to have a proper expressiveness for the purpose of inference and extensive software support at the stage of ontology edition, visualization, verification and further implementation in the system software components (in the Deployment phase).

In case of IOEM Formalization phase XML-based semantic languages, i.e.: RDF (Resource Description Framework) [14] and OWL will be used. They have appropriate expressiveness, are easy to use and are supported by available software development tools (e.g. Protégé, Jena library).

5.7 Deployment Phase

According to the metamodel presented in Figure 2, an ontology application can occur in the phase of system development or during the run-time. An ontology used in the development phase can be treated as the system development artifact. In case of an ontology application during the run-time, it can be considered as a software component that should be subject of verification and testing.

Decision on using OWL for formal ontology specification to some extent limits platforms that can be used during deployment. Currently, the most advanced tools and libraries have been developed for the Java platform, e.g. Jena, OWL API, Jess (as a rule engine). Thus, components using ontologies in the INSIGMA system are usually coded in Java; if an integration with another platforms is needed, a wrapper in form of web services is used.

5.8 Evaluation Phase

Evaluation is a process aimed at validating and verifying the ontology in terms of its scope, consistency and expressiveness. It relates the created ontology to the requirements defined in the Specification phase (possibility to answer competency questions, use cases coverage) but it lasts as long as the whole process of IOEM (see Fig.1). Evaluation relates therefore to all the products of subsequent phases and is carried out periodically even in parallel to the ontology lifecycle. Even though the methodology consists of steps, it is always possible to take a step back and make corrections that will make the ontology tailored to the needs of the project. Moreover, it is assumed that creating the complete set of classes, their relations and axioms within one course of the Conceptualization and Formalization phases is impossible. It is necessary to run at least 2 iterations after which the resulting model is evaluated against semantics, syntactic, coherence, coverage and, what is crucial, adherence to the competency questions and use cases.

In terms of the Specification phase, evaluation covers the aim and scope of the ontology and use cases. Initial conceptualization is verified against specification, mostly on the basis of review of the informal descriptions. The architecture of the ontology set in the Modularization phase is evaluated against the possibility of multiple usage of the components, possibility to work in parallel on their development and identifying possible limitations on implementation.

Evaluation process related to the Deployment phase employs techniques not strictly related to ontologies but regular mechanisms of software validation and verification.

5.9 Maintenance Phase

Maintenance is a process that corrects and updates the ontology. This phase is usually performed at the stage of ontology utilization in particular system components. Due to the fact, that the duration of the system development phase was established to 5 years, several software components will be successively created within this period. Particular domain ontologies will be though maintained by different task groups and in different timeframe. It is necessary for the ontology creator to actively supervise the ontology utilization and support the ontology maintenance. Although the Specification phase bases on the questionnaire that helps to develop the most appropriate assumptions, they may be not valid after the final system components are developed. The aim of the Maintenance phase is though to react on changes in system functionality, operation and implementation in order to provide the biggest benefit for the project.

In such a broad project as INSIGMA, particular domain ontologies are strongly related to their applications and deployment in particular components of the system. The changes included by a group of developers responsible by particular component must not negatively influence other modules and domain ontologies. It is very important though to assess the scope of ontology that supports only particular objective and parts of the ontology that have impact on many processes.

6 Summary

The paper presents IOEM, a methodology for ontology development elaborated for the INSIGMA project. Although prepared for a particular use, the methodology is quite general and can be used in a large variety of IT projects requiring ontology components. It is particularly suitable for large and geographically distributed software projects.

The methodology was proposed as a result of analyses of various approaches and known methodologies for building ontologies. The goal of IOEM methodology is to provide a defined process comprising phases, supporting tools and well determined results. The methodology implementation is a complex process including several activities that involve different experts and work groups. At present, a number of ontologies within the INSIGMA project are concurrently

built. In general, their progress ranges from phase 1 to 4. So far, no significant corrections in the methodology was required. At the end of the project, the authors plan to assess the overall methodology and formulate lessons learned.

Acknowledgments. Work has been co-financed by the European Regional Development Fund under the Innovative Economy Operational Programme, INSIGMA project no.POIG.01.01.02-00-062/09.

References

1. Guarino, N.: Formal Ontology and Information Systems. In: Proceedings of FOIS 1998, Trento, Italy (1998)
2. Sure, Y., Studer, R.: On-To-Knowledge Methodology - Final Version. On-To-Knowledge EU IST-1999-10132 (2002)
3. Noy, N.F., McGuinness, D.L.: Ontology Development 101: A Guide to Creating Your First Ontology. Stanford University, Stanford (2001), <http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html>
4. Uschold, M., King, M.: Towards a Methodology for Building Ontologies. In: Proceedings of IJCAI 1995 Workshop on Basic Ontological Issues in Knowledge Sharing (1995)
5. Gruninger, M., Fox, M.S.: Methodology for the Design and Evaluation of Ontologies. In: IJCAI 1995, Workshop on Basic Ontological Issues in Knowledge Sharing (1995)
6. Kotis, K., Vouros, G.A.: Human Centered Ontology Management with HCONE. In: Proceedings of the IJCAI 2003, Ontologies and Distributed Systems Workshop, CEUR-WS.org., vol. 71 (2003)
7. De Nicola, A., Missikoff, M., Navigli, R.: A proposal for a unified process for ontology building: UPON. In: Andersen, K.V., Debenham, J., Wagner, R. (eds.) DEXA 2005. LNCS, vol. 3588, pp. 655–664. Springer, Heidelberg (2005)
8. Holsapple, C.W., Joshi, K.: Handbook of Knowledge Management. In: A Knowledge Management Ontology, ch.6. Springer, Heidelberg (2003) ISBN 3-540-20005-3
9. Gómez-Pérez, A., Fernández-López, M., Juristo, N.: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In: Proceedings of Symposium on Ontological Engineering of AAAI. Spring Symposium Series, Stanford, pp. 33–40 (1997)
10. Booch, G., Rumbaugh, J., Jacobson, I.: Unified Modeling Language User Guide, 2nd edn. Addison-Wesley Professional, Reading (2005)
11. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax, W3C Recommendation (2009), <http://www.w3.org/TR/owl2-syntax/#Imports>
12. IBM Certified Solution Designer - IBM Rational Unified Process V7.0, <http://www-03.ibm.com/certify/certs/38008003.shtml>
13. Leffingwell, D., Widrig, D.: Zarzadzanie wymaganiami. WNT Warszawa (2003)
14. Klyne, G., Carroll, J.J.: Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation (2004), <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>

A Framework for Building Logical Schema and Query Decomposition in Data Warehouse Federations

Rafał Kern, Krzysztof Ryk, and Ngoc Thanh Nguyen

Wroclaw University of Technology, Wyb. Wyspińskiego 27, 50-370 Wroclaw, Poland
Rafal.Kern@pwr.wroc.pl, 157657@student.pwr.wroc.pl,
Ngoc-Thanh.Nguyen@pwr.edu.pl

Abstract. In this paper a framework for building logical schema and query decomposition procedure for federations of data warehouses has been proposed. The logical schema of a data warehouse federation arises as the result of integration of the schemas of component data warehouses. A user query refers to the schema of the federation, therefore, the query decomposition process is needed.

Keywords: warehouse federation, schema integration, SQL query decomposition.

1 Introduction

A data warehouse most often treats the real world in a very narrow scope. If a company has several warehouses for which the scopes may overlap, for example, product sale, then for the same query, for example referring to the sale trend, the results could be different for different warehouses. If one wants to have a general view of the activities of the whole company then it is needed to integrate the warehouses in some way. One of possible approaches is to create a federation of them.

As a federation of warehouses we understand a system consisting of several warehouses and a federation management system, which contains the following elements:

- An integration procedure of the schemas of the component warehouses giving the logical schema of the federation.
- A query language for user who does not need to know the schemas of the component warehouses.
- A procedure which enables decomposition of user queries to the federation into sub-queries which are sent to the component warehouses.
- A procedure for integrating the answers gathered from the component warehouses.

Owing to this architecture a user can present his query to the federation as a whole and the answer he receives is a result of those integrated from the component warehouses. The requirements mentioned above suggest some similarity in organizing data warehouse federations and distributed databases. However, the architecture of a warehouse federation should differ from the architecture of a distributed database in two aspects: First, there is a very strict rule for data distribution in distributed databases (horizontal or vertical sections), and, as the result, a logical table in a distributed

database is a sum or join of some sections). In data warehouse federation, the rule is not so restrict. Secondly, although in distributed databases there is also the decomposition process for queries, however, there is no conflict in integrating answers from the sub-queries, since each sub-answers refer to different subjects. In data warehouse federation several data warehouses may refer to the same subject, and the several sub-answers may have overlapped subjects, which shall cause conflict solving in the integration task [10].

In this paper we present a formal approach for building the data warehouse federation schema and a decomposition procedure for queries. The schema of data warehouse federation arises as the result of integrating the schemas of the component data warehouses. The remaining part of this paper is organized as follows: Section 2 contains an analysis of related works. Section 3 includes the definitions of data warehouse federation and its notions. A procedure for integrating data warehouse schemas is presented in Section 4. Section 5 presents a SQL query decomposition algorithm. Some implementation aspects are included in Section 6.

2 Related Works

In the literature it is hard to find formal definitions of federated data warehouses. However, the description of data warehouse federations appearing in different papers allows formulating an informal definition of this kind of databases. The database federation as a form of organization proposed by Sheth and Larson [1] seems to be good pattern for defining data warehouses federations described in [3], [4], [7], [11]. According to it a set of data warehouses can be called a federation when it fulfills the conditions of autonomy, heterogeneity and dispersion. The components participating in the federation have strong autonomy and may have very different structures. In another work Heimbigner and McLeod [2] defined a federation of data warehouses as a set of components which are communicating one with another by rules stored in the federal dictionary.

It is assumed that a data warehouse federation works on logical level. This means that none computations on physical data are made. A request from external users/applications are decomposed and forwarded to suitable component data warehouses. After then sub-queries are handled the responses are merged and sent back to the user.

By the autonomy of federation components we understand the fact that its regular performance should not be strongly affected by joining the federation. Sheth and Larson [1] proposed four levels of autonomy: design, communication, execution and association. Each component may be developed independently. Also on components level the decisions if, and on which rules the communication with other components should be performed, are made. Moreover, each of them decides autonomously which parts of data will be shared with the federation. The most important aspect of autonomy is the free choice to joining and leaving the federation. Each component data warehouse must fulfill initial assumptions and regulation taken into account during its development. These can be for example: the purpose, the law regulation, participation in other federations etc. One data warehouse may theoretically join unlimited number of federations. The heterogeneity feature of means, that we cannot made any

assumptions about structural similarities. This refers to the technology, inner structure and presentations layer.

In federation architecture two main layers can be differed: the component layer [5] and the federation layer (sometimes called also the coordinator layer [1], [3]). The coordinator layer is responsible for maintenance of the structure of federation and the common dimensional schema in tightly coupled approach. It constitutes a transparent interface for external users hiding components heterogeneity [1], [3], [9]. In loosely coupled approach the component maintain the federation and the coordinator layer is reduced to minimum [2]. In the coordinator layer two more levels can be noticed: external schema and federation schema. In the component layer there occur export, component and local schemas.

External schema defines the communication between federation and external applications. It is responsible for federation's adaptation to the user requirements, introducing additional integrity constraints between components and controls the access to data stored in federation. Each federation may have more than one external schema for each user class. By user class we mean a set of users performing similar actions on federation [1]. This schema also contains information how obtain the processed query results.

The federation schema is a result of export schemas integration. Here takes place the integration of dimensions and facts obtained through the import schemas. It is possible to store more than one federation schema. In loosely coupled approach the import schemas are stored in every component [2]. In tightly coupled – in federated schema [1], [3]. Mappings between components schemas and the global schema are stored in data dictionary which is part of federation layer. In [12] some modeling approaches are discussed.

The component warehouse is a structure build on native, local data warehouse schema. It is extended by the component schema and export schema. The component schema is in fact the local schema described by expressions using the common dimensional model. In the export schema each component determines which data it will share with other federation participants. The communication between components uses the federation, export and component schemas. In the last one it is possible to add some metadata enriching the knowledge about ties between components.

In [1] an additional, auxiliary schema was proposed. It stores data which is unavailable in any of the origin data warehouse but is very useful in query optimization or incompatibility solution. The data may be used by mediator [5] and may be obtained during ETL processes [6].

According to [2] every component should have own communication schemas with each of other components participating in federation. Unfortunately, every time new participants join federation all the "old" members must update its communication schemas. On the other hand, in [1] a common dimensional model was proposed. This approach assumes that each component must transform its own schema into one, common model. By such a solution the "old" participants are invulnerable to the federation growth.

The common dimensional model must be worked out as a result of many transformations of the component schemas. In loosely coupled approach it is defined as the largest common part of the local schemas [7], [8]. In tightly coupled approach – as a combination of the origin data warehouses schemas.

Most of existing solutions treat federation as a common view on some business area, e.g. sales. The common view was defined as “largest common schema” in [7]. In [1] - as “*a data model to which schemas of different components are translated for the purpose of representation in a common format*”. Sometimes it may be very hard to find a common model for several different schemas. Structures which are used for queries-processing optimization may make the integration process very expensive and long. It does not support full cross-functional and cross-domain queries. For example, formulating queries about difference between production costs and sales income for some certain market area would be impossible to handle. Therefore, sometimes it is much easier to use more than one schema for more accurate description of the business areas.

3 Basic Notions

Assume that there are given data warehouses H_1, H_2, \dots, H_n . Each H_i (for $j = 1, 2, \dots, n$) can be represented by a tuple

$$H_i = (D_0^i, D_1^i, D_2^i, \dots, D_{ai}^i)$$

where D_0^i is the fact table, D_j^i is a dimension for $j = 1, 2, \dots, ai$. Between a dimension D_j^i and the fact table there occurs a relationship of type $1 - \infty$, that is denoted by symbol $D_j^i \rightarrow D_0^i$.

A federation F consisting of data warehouses H_1, H_2, \dots, H_n is denoted by

$$F = (D_0, D_1, D_2, \dots, D_m)$$

where D_0 is the fact table, D_j is a dimension for $j = 1, 2, \dots, m$. Between a dimension D_j and the fact table there occurs a relationship of type $1 - \infty$, that is denoted by symbol $D_j \rightarrow D_0$.

Each attribute occurring in a dimension or measure occurring in the fact table of the federation should come from one or more component data warehouses. Therefore, we denote an attribute a in a dimension of the federation by the following parameters:

- Attribute name in the federation,
- A list of pairs $(D_j^i, name)$ where D_j^i identifies the data warehouse and the dimension in it the attribute comes from; $name$ denotes its name in this dimension.

Similarly, a measure occurring in the fact table of the federation is characterized by the following parameters:

- Measure name in the federation,
- A list of pairs $(D_0^i, name)$ where D_0^i identifies the data warehouse and the fact table in it the measure comes from; $name$ denotes its name in this fact table.

4 Building Federation Schema Using Integration Methods

In this preliminary algorithm we work only with data warehouses with star schema, so we do not deal with hierarchies in dimensions.

Input:

H_p – data warehouse schema defined as $H_i = (D_0^i, D_1^i, D_2^i, \dots, D_{\alpha i}^i)$

F – existing federation schema defined as $F = (D_0, D_1, D_2, \dots, D_m)$

Output:

F – federation schema defined as $F = (D_0, D_1, D_2, \dots, D_m)$ after integration with H_p

We will use following notations:

- a_name – name of attribute a
- b_name – name of measure b
- $D_x \sim D_z$ – D_x is similar to D_z (expert's decision)
- $a_x \equiv a_z$ – similar attributes (expert's decision)
- $b_x \diamond b_z$ – similar measures (expert's decision)

The algorithm works iteratively. In the first iteration the input federation schema looks as follows:

$$F = D_o \text{ where } D_o = \emptyset.$$

In each iteration a data warehouse schema is being integrated with federation schema. It consists of two steps: measures integration and dimension integration. If input data warehouse does not have any measures, the first step may be skipped.

The general idea is as follows:

1. For each measure from input data warehouse try to find corresponding measure in federation schema.
 - a. If such a measure exists in federation schema add a mapping between them.
 - b. If none of the federation measures corresponds to the current one add it to the federation and make a mapping between new measure and the current one.
2. For each dimension from input data warehouse look for a corresponding dimension in federation schema
 - a. If such a dimension exists, check the attributes matching. If any of the current dimension's attribute does not has equivalent in matching dimension extend it by this attribute. In both cases add a mapping between these dimensions.
 - b. If no matching dimension was found, add the current dimension to the federation schema.

Measures integration

```

foreach measure  $b'$  in  $D_0^p$ 
  if  $\exists b \in D_0: b \diamond b' \wedge b$  is characterized by a tuple  $(b\_name, list)$ 
    list = list  $\cup \{(D_0^p, b\_name)\}$ 
  else
     $D_0 = D_0 \cup \{(b\_name, \{(D_0^p, b\_name)\})\}$ 
  endif
endforeach

```

Dimensions integration

```

foreach  $D_y^p$  in  $H_p$ ,  $y = 1, 2, \dots, \alpha_p$ 
  if  $\exists D_t \in F: D_t \sim D_y^p$ 
    foreach attribute  $a'$  in  $D_y^p$ 
      if  $\exists a \in D_t: a \equiv a' \wedge a$  is characterized by  $(a\_name, list)$ 
        list = list  $\cup \{(D_y^p, a'\_name)\}$ 
      else
         $D_t = D_t \cup \{(a'\_name, \{(D_y^p, a'\_name)\})\}$ 
    endif
  endforeach
else
   $D_t = \emptyset$ 
  foreach  $a''$  in  $D_y^p$ 
     $D_t = D_t \cup \{(a''\_name, \{(D_y^p, a''\_name)\})\}$ 
  endforeach
   $F = F \cup \{D_t\}$ 
endif
endforeach

```

For simplification we assume that each dimension in federation has a reference in the fact table so it is not given directly in this algorithm.

Example 1

Let us consider two schemas related to sales, illustrated on Figures 1 and 2.

H_1 – InternetSales

$H_1 = (D_0^1, D_1^1, D_2^1, D_3^1)$

Fact table:

$D_0^1 = \{TransactionId, TransactionDate, ProductId, IndividualClientId, Value\}$

Dimensions:

1. Date: $D_1^1 = \{Id, Year, Quarter, Month\}$
2. Product: $D_2^1 = \{ProductId, Color, Size, Model\}$
3. IndividualClient:

$D_3^1 = \{ClientId, Email, Firstname, Lastname, Country, BirthYear\}$

H_2 – SalesWithSalesRepresentative

$H_2 = (D_0^2, D_1^2, D_2^2, D_3^2, D_4^2)$

Fact table:

$D_0^2 = \{TransactionId, TransactionDate, ProductId, CompanyClientId, Value\}$

Dimensions:

1. Date: $D_1^2 = \{Id, Year, Quarter, Month\}$
2. Product: $D_2^2 = \{ProductId, Color, Size, Model\}$
3. CompanyClient: $D_3^2 = \{ClientId, NIN, Address, Country\}$
4. SalesRepresentative: $D_4^2 = \{SalesRepresentativeId, Firstname, Lastname, Fee, BirthYear, BirthMonth, BirthDay\}$

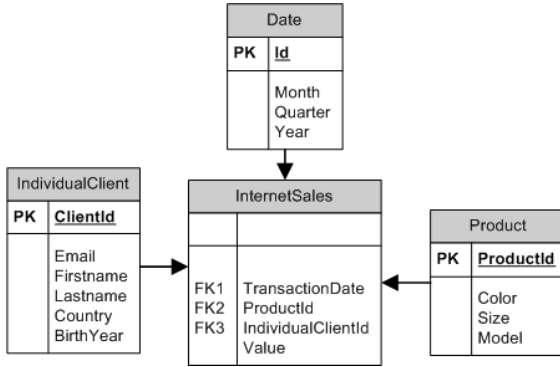


Fig. 1. An example schema for internet shopping

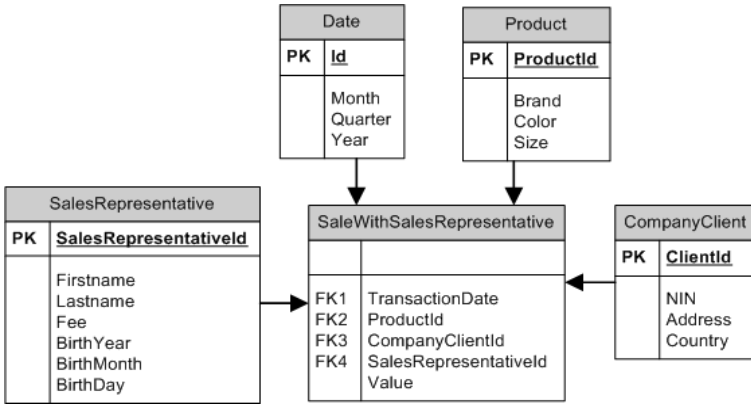


Fig. 2. An example schema for sales with a representative

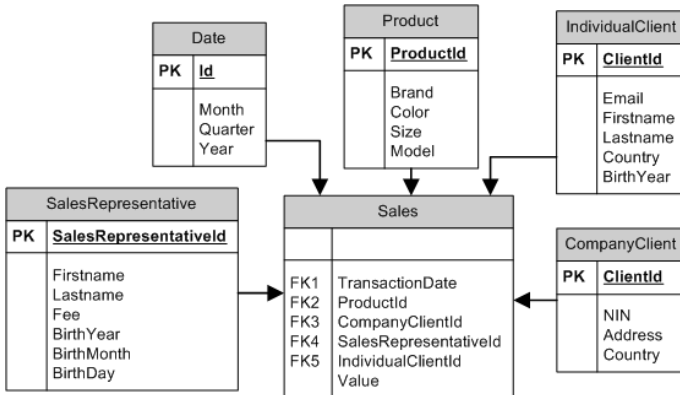


Fig. 3. Result of schemas integration from Figures 1 and 2

The algorithm takes two iterations and starts with following conditions:

$$F = (D_o), \quad \text{where } D_o = \emptyset$$

In the first iteration H_1 is being integrated with F . After that, the federation F has following structure:

$F = (D_0, D_1, D_2, D_3)$ where:

$$D_0 = \{(Value, \{(D_0^1, Value)\}), referencesToDimensions\}$$

$$D_1 = \{(Id, \{(D_1^1, Id)\}), \\ (Year, \{(D_1^1, Year)\}), \\ (Quarter, \{(D_1^1, Quarter)\}), \\ (Month, \{(D_1^1, Month)\})\}$$

$$D_2 = \{(ProductId, \{(D_2^1, ProductId)\}), \\ (Color, \{(D_2^1, Color)\}), \\ (Size, \{(D_2^1, Size)\}), \\ (Model, \{(D_2^1, Model)\})\}$$

$$D_3 = \{(ClientId, \{(D_3^1, ClientId)\}), \\ (Email, \{(D_3^1, Email)\}), \\ (Firstname, \{(D_3^1, Firstname)\}), \\ (Lastname, \{(D_3^1, Lastname)\}), \\ (Country, \{(D_3^1, Country)\}), \\ (BirthYear, \{(D_3^1, BirthYear)\})\}$$

In the second iteration the H_2 schema will be integrated with the result of the first iteration.

1. It is easy to notice, that the measure Value from H_2 corresponds to measure from federation. So:

$$D_0 = \{(Value, \{(D_0^1, Value), (D_0^2, Value)\}), referencesToDimensions\}$$

2. The D_1^2 (Date dimension) fits D_1 so D_1 should be extended by additional mapping. Thus, we have:

$$D_1 = \{(Id, \{(D_1^1, Id), (D_1^2, Id)\}), \\ (Year, \{(D_1^1, Year), (D_1^2, Year)\}), \\ (Quarter, \{(D_1^1, Quarter), (D_1^2, Quarter)\}), \\ (Month, \{(D_1^1, Month), (D_1^2, Month)\})\}$$

3. The D_2^2 (Product dimension) corresponds to D_2 . However, some attributes are unique for both dimensions. According to mentioned algorithm D_2 should also be extended by additional mapping and new attributes should be added. Finally, as a result we gain:

$$D_2 = \{(ProductId, \{(D_2^1, ProductId), (D_2^2, ProductId)\}), \\ (Color, \{(D_2^1, Color), (D_2^2, Color)\}), \\ (Size, \{(D_2^1, Size), (D_2^2, Size)\}), \\ (Model, \{(D_2^1, Model)\}), \\ (Brand, \{(D_2^2, Brand)\})\}$$

4. The D_3^2 (Company Client dimension) does not fit any of existing dimensions in federation, therefore it should be added to the federation.

$$D_4 = \{(ClientId, \{(D_3^2, ClientId)\}),$$

- $(NIN, \{D_3^2, NIN\})$,
 $(Address, \{D_3^2, Address\})$,
 $(Country, \{D_3^2, Country\})$
5. Similarly, the D_2^4 (Sales Representative dimension) does not fit any of existing dimensions so it should also be added as a new one into federation.
 $D_5 = \{(SalesRepresentativeId, \{D_4^2, SalesRepresentativeId\})$,
 $(Firstname, \{D_4^2, Firstname\})$,
 $(Lastname, \{D_4^2, Lastname\})$,
 $(Fee, \{D_4^2, Fee\})$,
 $(BirthYear, \{D_4^2, BirthYear\})$,
 $(BirthMonth, \{D_4^2, BirthMonth\})$,
 $(BirthDay, \{D_4^2, BirthDay\})\}$
 6. Finally, we gain new federation schema (Figure 3), which is the foundation of decomposition process performed in Section 5.
 $F = (D_0, D_1, D_2, D_3, D_4, D_5)$

5 Algorithm for SQL Query Decomposition

Queries are integral and essential part of the federation. Due to the federated data warehouse assumptions, queries are quite specific – they are being prepared by user and dedicated for the integrated federation schema. One of the federation management system purposes is to translate query into sub-queries, which will be sent to component warehouses. According to this fact, algorithm for SQL Query Decomposition must be introduced. Because of lack of space only an example is given here. The algorithm will be described broader in another paper.

Example 2

User wants to ask the following query to the federation:

```

SELECT Brand, Color
FROM Product INNER JOIN Sales ON Product.ProductId = Sales.ProductId
WHERE Size > 38
GROUP BY Color;

```

The following part-statements can be extracted from original query:

```

SELECT statements: {Brand, Color}
FROM statements: {Product INNER JOIN Sales ON
                  Product.ProductId = Sales.ProductId},
WHERE statements: {Size > 38}
GROUP BY statements: {Color}

```

Note: Each of the FROM statements consists of: table names, join operation and join condition; Each of WHERE statements consists of: attribute name, condition, scalar (or another attribute name).

According to the presented algorithm, in the first step we should check whether mappings for each FROM statement exist in component data warehouse. In other words, our purpose is to check, if each element from $\{Product.ProductId,$

Sales.ProductId} has mapping in a component data warehouse. For the first component data warehouse, mapping for each element from presented set, exists – FROM statement for this warehouse can be build: *{Product INNER JOIN SaleWithSalesRepresentative ON Product.ProductId = SaleWithSalesRepresentative.ProductId}*.

Next step is to check mappings for each of the SELECT statements: *{Brand, Color}*. Similarly to the previous step, purpose is to check whether exists mappings for ‘Brand’ and ‘Color’ in component data warehouse. For the first data warehouse, mapping exists for both attributes, in consequence SELECT statements can be built for this component warehouse: *{Brand, Color}*.

For WHERE statements, situation is similar to the previous – mapping for each element of *{Size > 38}* exists, and WHERE statements for this warehouse can be build.

Last step is to check GROUP BY statements: mapping exists for ‘Color’ in component data warehouse. Based on constructed sub-statements, SQL statement can be constructed for this data warehouse:

```
SELECT Brand, Color
FROM Product INNER JOIN SaleWithSalesRepresentative ON
    Product.ProductId = SaleWithSalesRepresentative.ProductId
WHERE Size > 38
GROUP BY Color;
```

For the second data warehouse course of the algorithm is analogous, with one difference: mapping for one of SELECT statement (‘Brand’) does not exists – according to the algorithm, it is possible to query this component warehouse (this attribute will be excluded in query). Based on constructed sub-statements, SQL statement can be constructed for this data warehouse:

```
SELECT Color
FROM Product INNER JOIN InternetSales ON
    Product.ProductId = InternetSales.ProductId WHERE Size > 38
GROUP BY Color;
```

6 Implementation

Algorithm for SQL Query Decomposition was implemented in Java programming language. First package (*pl.pwr.krn.domain*) contains classes which are: object representation of SQL Query (for example: *SQLQuery.java*, *FromStatement.java*, *SelectStatement.java*, *JoinOperationEnum.java* etc.), representation of attribute mapping (*DWMMapping.java*). Second package (*pl.pwr.krn.extractor*) consists of *QueryExtractor.java*, which is responsible for translating query to the component data warehouse. Next package (*pl.pwr.krn.main*) contains main class of the program – *SQLDeAl.java*. Penultimate package (*pl.pwr.krn.test*) consists of tests (written in *jUnit*) dedicated for this application. Last package (*pl.pwr.krn.util*) contains utility classes, which are being used as helpers, classes are: *DWMMappingUtil.java* (responsible for searching attribute mappings in component data warehouses) and *QueryBuilder.java* (responsible for building SQL query, based on its object representation).

Our assumption was that input query has a specific structure - user defines further statements (SELECT, FROM, ...). Based on this specific query structure, SQL query object representation is build. Next step is to translate given query, to component data warehouses. QueryExtractor.java checks whether each component part of each statement has a mapping in specific data warehouse (according to presented algorithm). During this process SQLQuery.java is being built. After this, QueryBuilder.java translates "object" SQL query, to SQL query language.

7 Conclusions

In this paper a framework for building logical schema and query decomposition procedure for federations of data warehouses has been proposed. The results presented in this paper can be a foundation for a formal model for data warehouse federations. The future works should concern the implementation of the approach and working out a mechanism for integration results given by component warehouses.

Acknowledgments. This research was partially supported by European Union within European Social Fund under the fellowship and by Polish Ministry of Science and Higher Education under grant no. N N519 407437 (2009-2012).

References

1. Sheth, A.P., Larson, J.A.: Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Comput. Surv.* 22(3), 183–236 (1990)
2. Heimbigner, D., McLeod, D.: A federated architecture for information management. *ACM Trans. off. Znf. Syst.* 3(3), 253–278 (1985)
3. Berger, S., Schrefl, M.: From federated databases to a federated data warehouse system. In: *HICSS 2008: 41st Annual Hawaii Intl. Conf. on System Sciences*, pp. 394–394. IEEE Computer Society, Los Alamitos (2008)
4. Banek, M., Tjoa, A.M., Stolba, N.: Integrating Different Grain Levels in a Medical Data Warehouse Federation. In: Tjoa, A.M., Trujillo, J. (eds.) *DaWaK 2006*. LNCS, vol. 4081, pp. 185–194. Springer, Heidelberg (2006)
5. Akinde, M.O., et al.: Efficient OLAP Query Processing in Distributed Data Warehouses. *Inf. Syst.* 28(1-2), 111–135 (2003)
6. Jindal, R., Acharya, A.: Federated Data Warehouse Architecture. White paper, Wipro Technologies (2003)
7. Schneider, M.: Integrated vision of federated data warehouses. In: *Proceedings of International Workshop on Data Integration and Semantic Web, CEUR-WS (DisWeb 2006)*, vol. 238, pp. 336–347 (2006)
8. Cabibbo, L., Torlone, R.: Integrating Heterogeneous Multidimensional Databases. In: *Proceedings of SSDBM 2005*, pp. 205–214 (2005)
9. Litwin, W., Mark, L., Roussopoulos, N.: Interoperability of Multiple Autonomous Databases. *ACM Comput. Surv.* 22(3), 267–293 (1990)
10. Nguyen, N.T.: *Advanced Information and Knowledge Processing*. Springer, London (2008)
11. Nemati, H.R., Steiger, D.M., Iyer, M.S., Herschel, R.T.: Knowledge warehouse: an architectural integration of knowledge management, decision support, artificial intelligence and data warehousing. *Decision Support Systems* 33, 143–161 (2002)
12. Lenzerini, M.: Data Integration: a Theoretical Perspective. In: *Proceedings of PODS 2002*, pp. 233–246. ACM, New York (2002)

A Distance Function for Ontology Concepts Using Extension of Attributes' Semantics

Marcin Pietranik and Ngoc Thanh Nguyen

Institute of Informatics, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370, Wrocław, Poland
{marcin.pietranik,ngoc-thanh.nguyen}@pwr.wroc.pl

Abstract. This paper contains presentation of our work on creating robust methodology of ontology alignment. After detailed analysis of literature we have noticed that former approaches to this task covered only the surface of the issue, skipping the possible potential of including the semantics of basic building blocks of ontologies (which are concepts' attributes). We have noticed that attributes can alter their meanings when incorporated within different concepts and giving them explicit semantics, can be utilized as criterion for identifying correspondences between them. We claim that such holistic approach can improve the reliability of yielded results and eventually move the focus from finding alignments based on labels of concepts to mapping their semantic content.

Keywords: ontologies, ontology alignment, knowledge management.

1 Introduction

Ontologies are frequently used as a foundation of many modern computer systems that provide some kind of knowledge base along with functionality of gathering its content, modifying and providing inference mechanisms based on facts included in it.

Distributed environments also often require the automated method of transforming information from one knowledge base into another. Therefore, the task of finding valid ontology mappings appears. Recently developed approaches concentrated only on the surface of this task, providing solutions based on concepts' labels and designating symmetrical results, which were based on pairwise correspondences between ontologies. Such simplification is easy to implement and visualize, but omits the grounding level of the issue, causing close attachment to representing ontologies with OWL standard and fixed benchmarks.

In this paper we extend our work that has been presented in previous publications [10] and [11], where we concentrated on defining concepts' structures built on top of combination of assigned attributes and semantics they acquire. The main idea was how intuitively seen correspondences between attributes can influence the whole process of aligning ontologies. After careful analysis we have defined formal criteria of such characteristics. We have also noticed that previous methods did not consider the way that attributes may change when included in

different concepts, lacking reflection about possible ambiguity of calculated results. Solutions based only on varying naming convention to avoid such problems are, in our opinion, insufficient.

Developing further these observations, we have formulated the algorithm that analysis concepts' structures, identifies correspondences between them by utilizing formal criteria and eventually calculates truly semantic distance between two concepts.

To illustrate our ideas about approaching ontology alignment, throughout this paper we will adopt the ensuing example. Let the two concepts be defined as follows. These concepts are members of two separate ontologies describing

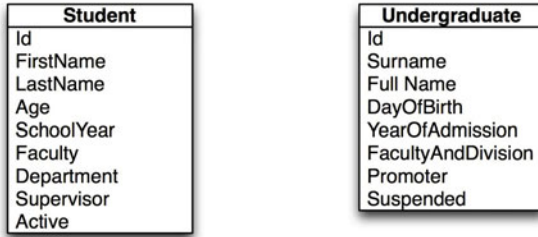


Fig. 1. Example classes structures

organizational charts of different universities, that plan to exchange students from various faculties or school years. Therefore, the need of transforming the method of storing personal data appears.

Our paper has a following structure. Section 2 contains overview of basic notions used throughout the whole paper. Part 3 gives brief overview of the work that has been done in the topic of attribute-based ontology alignment that can be found in literature. Section 4 contains preliminary assumptions that we have accepted, along with postulates for reliable concept distance algorithm and finally our algorithm itself. The last section provides summary and short overview of ideas for future works.

2 Basic Notions

Treating [9] as a starting point we define ontology as a triple:

$$O = (C, R, I) \tag{1}$$

where C is the set of concepts, R is a set of relationships between them (defined as $R \subseteq C \times C$) and I is a set of instances.

We define concept c from set C also as a triple:

$$c = (Id^c, A^c, V^c) \tag{2}$$

in which Id^c denotes a concept label (an informal name of a concept), A^c is a set of attributes belonging to the particular concept and V^c is a set of domains of attributes from A^c . The triple c is called *concept's structure*.

We assume existence of finite set S containing atomic descriptions of attributes' semantics. An element from set S is a basic description given in natural language. L_s is the formal language, incorporating symbols from S and basic logic operators \neg, \vee, \wedge . L_s is a sublanguage of the sentence calculus language.

Definition 1. *By semantics of attributes within concepts we call a partial function:*

$$S_A : A \times C \rightarrow L_s \tag{3}$$

Logic sentences are being assigned to attributes incorporated in concepts. This approach allows us to give different semantics to the same attribute - imagine the attribute *Address* included both in the concept *Person* and the concept *Webpage*. Obviously these two assignments give heterogenous meanings to the same attribute.

Definition 2. *By concepts' semantics we define a function:*

$$S_C : C \rightarrow L_s \tag{4}$$

This function assigns semantical descriptions to concepts and will further allow us to precisely distinguish concepts that contain the same or very similar structures. Such approach to expressing semantics of attributes and concepts allow providing unequivocal criteria for identification of correspondences between concepts' attributes and concepts. We define them as follows:

Definition 3. *Two attributes $a, b \in A$ are equivalent referring to their semantics (semantical equivalence) if the formula $S_A(a, c_i) \Leftrightarrow S_A(b, c_j)$ is a tautology for any two $c_i, c_j \in C(c_i \neq c_j)$. To mark this relation we will use the symbol \equiv .*

For example, the attribute *LastName* is equivalent to attribute *Surname* and we can denote this fact as $LastName \equiv Surname$.

Definition 4. *The attribute $a \in A$ in concept $c_i \in C$ is more general than attribute $b \in A$ in concept $c_j \in C$ referring to their semantics (semantical generality) if the formula $S_A(b, c_j) \Rightarrow S_A(a, c_i)$ is a tautology for any two $c_i, c_j \in C(c_i \neq c_j)$. To mark this relation we will use the symbol \uparrow .*

For example, the attribute *Age* is more general than attribute *DateOfBirth*, because knowing the date of birth of some person, we can easily calculate his age, but knowing his age we cannot designate his accurate date of birth. We denote this fact as $DateOfBirth \uparrow Age$.

Definition 5. *Two attributes $a, b \in A$ are in contradiction referring to their semantics (semantical contradiction) if the formula $\neg(S_A(a, c_i) \wedge S_A(b, c_j))$ is a tautology for any two $c_i, c_j \in C(c_i \neq c_j)$. To mark this relation we will use the symbol \downarrow .*

For example, two attributes *IsActive* and *IsSuspended* are in contradiction and we denote this fact as $IsActive \downarrow IsSuspended$.

In our example from Figure 1 for simplicity we assume that the semantics of attributes are given as conjunctions of atoms or their negations and the distance function between them is a normalized Levenshtein Edit Distance. Having the set of atomic descriptions (where in brackets we give abbreviations of it's elements) $S = \{Name(a), Surname(b), Age(c), DateOfBirth(D), YearOfAdmission(e), SchoolYear(f), Faculty(g), Department(h), TeacherName(i), Supervisor(j), ActiceStatus(k), Id(l)\}$ we define semantics of attributes within concepts in the table below:

Table 1. Attributes' Semantics

Attribute's Name	Semantics
Active	k
Age	c
DayOfBirth	$c \wedge d$
Department	h
DivisionAndFaculty	$g \wedge h$
Faculty	g
FirstName	a
FullName	$a \wedge b$
Id	l
LastName	b
Promoter	$i \wedge j$
SchoolYear	f
Supervisor	$i \wedge j$
Surname	b
Suspended	$\neg k$
YearOfAdmission	$e \wedge f$

Incorporating criteria from Definitions 3, 4 and 5 we can identify following relationships: $\{LastName \equiv Surname, Supervisor \equiv Promoter, DayOfBirth \uparrow Age, DivisionAndFaculty \uparrow Department, DivisionAndFaculty \uparrow Faculty, FullName \uparrow FirstName, FullName \uparrow LastName, FullName \uparrow Surname, YearOfAdmission \uparrow SchoolYear, Active \downarrow Suspended\}$.

3 Related Works

Attribute matching is an issue associated more frequently with issues related to database schema matching. Despite a little interest within work on ontology alignment, it can be considered as a basic step in finding reliable mappings.

The comprehensive description of this topic in the context of ontology alignment can be found in [3]. It introduces the concept of the *model of attribute correspondence* that contains a finite set of attributes along with the similarity matrix that can be utilized as a filter of valid mappings. Included division

of different approaches to calculating this values contains the Term-based and value-based solutions. It is based on the work that has been done on ontology alignment and Semantic Web (further described in [2] or [12]). It also discusses the classification of possible correspondences, which are being divided into *contextual*, *semantic* and *probabilistic*.

In [4] the set of possible relationships that can occur between attributes is introduced. Authors identify connections that are similar to our ideas: *equivalence*, *subsumption*, *disjointness* and *unknown*. Based on this model, authors approach the task of designating mappings between attributes by automated discovery of mentioned relationships and incorporating their meaning. The main disadvantage of described solution is the fact that finding connections between attributes is build around tokenization and lemmatization of their labels and then utilization of external thesaurus, such as Wordnet (<http://wordnet.princeton.edu/>). What is interesting authors also provide so called *lattice of correspondences*, which describes strengths of found relationships (for example, claiming that *equivalence* is stronger than *subsumption*). An efficient algorithm that designates attribute correspondences is provided, but no formal conditions describing members of *lattice of correspondences* and the way they influence the whole can be found.

These ideas are further developed in [8] which introduce incorporation of machine learning techniques that increases the ability of choosing methods of designating matchers.

An interesting methods of analyzing the semantics of attributes can be found in [7]. Authors give formal conditions of identifying relationships between heterogeneous attributes with regard to their domains and valid valuations. The extension of this model into the model of *uncertain semantic relationship* is also provided along with illustrative application examples.

The topic of *ontology alignment* is accurately described in [2], which include detailed investigation of both modern and former solutions. It contains mainly different approaches to calculating similarities between elements of ontologies and the variety of evaluation methods of obtained results.

4 Distance Function for Ontology Concepts

4.1 Preliminaries

Following definitions described in Section 2 of this paper, we wanted to approach the problem of aligning ontology concepts with regard to all of their features. Therefore, we have distinguished three layers of abstraction in which we have incorporated valuable information about concepts, their structures and the way they influence the process of finding valid mappings.

1. *Distance between concept's semantics*
2. *Distance between concept's structures*
3. *Partial alignments between concept's attributes*

Developing our previous work described [11] in the first layer we wanted to incorporate flexible method of calculating distances between logic statements to unequivocally state how close two concepts are referring to their semantics. In other words, we wanted to come up with a method of calculating the strength of a possible relationship that can connect two concepts and by *semantical distance* we call a function defined as $d_s : L_s \times L_s \rightarrow [0, 1]$. A few different methods of particular semantical distance functions were described both in [10] and [11]. Due to the limited space available for this paper, in the further parts we will assume the existence of functions designated as d_A and d_C that calculate distances between semantics of attributes and concepts.

In the second layer of our model we have incorporated the way of comparing structures of concepts - their sets of attributes and their semantics. In this stage we have also been able to identify the need of including not only comparing raw sets of attributes, but also incorporating possible relationships that can occur between them. The initial inspiration came from the algorithm for integrating ontologies on the concept level from [9]. More detailed description of this layer along with the actual algorithm can be found in next section of this paper.

The third layer came from the idea that there are situations in which, despite the high structure distance, we still are able to transform some data from one concept to another. For example, imagine two concepts *Employee* and *Worker*. The first one contains attributes (*name, position, age, employment_date*), while the structure of the second has (*id, job, payment*). Obviously the structure distance is high, but still the *equivalence* relationship holds between attributes *position* and *job*. Analyzing this example, made us thinking that there should be a method of handling such issues and providing the way of aligning only fragments of concepts. Eventually we have developed a *partial alignment* defined as follows:

$$Align_p(c_1, c_2) = \{(a, b, \phi) : a \in A^{c_1}, b \in A^{c_2}, \phi \in \{\equiv, \uparrow, \downarrow\} \wedge (a\phi b \vee b\phi a)\} \quad (5)$$

where A^{c_1} and A^{c_2} are sets of attributes of concepts $c_1 = (Id_1, A^{c_1}, V^{c_1})$ and $c_2 = (Id_2, A^{c_2}, V^{c_2})$ from two ontologies O_1, O_2 . This structure contains triples of attributes from concepts along with the relationship that occurs between them. Such solution provides the way of transforming only portions of information, becoming independent from fixed threshold distance filtering.

4.2 Postulates for Concept Distance

In this paper we concentrate on presenting the algorithm that calculates the distance function between two concepts. Therefore, its signature can be written as $d_C : C \times C \rightarrow [0, 1]$. The requirements for this function are expressed in the following postulates.

1. The function is not *symmetrical*, yet it is also not *anti-symmetrical*.
2. For two concepts c_1, c_2 where $A^{c_1} = \{a\}$, $A^{c_2} = \{b\}$ and $a \uparrow b$ the function should satisfy the condition $d_C(c_1, c_2) < d_C(c_2, c_1)$

3. If in one of the concepts the set of attributes contains much more general attributes, and the second has more detailed attributes, the distance from the less general concept to more detailed should be lower than other way around.

The first postulate illustrates the natural approach to transforming concepts, where the symmetry requirement is too strict - in other words, there are situations in which mapping from one concept to another is simpler than from the second to the first. What is more, the method of transforming one concept is not necessarily identical as the method of transforming the second concept. This condition is further developed in Postulate 2 and 3 that express the intuition that having more detailed information simplifies mapping them into more general knowledge.

The next section contains the description of our algorithm that satisfy given postulates along with example of acquired results.

4.3 Algorithm for Determining Concept Distance

We assume the existence of two different ontologies $O_1 = (C_1, R_1, I_1)$ and $O_2 = (C_2, R_2, I_2)$ consistent with Equation 1. Subsequently, we assume that within these ontologies exist two concepts $c_1 = (Id_1, A^{c_1}, V^{c_1})$ and $c_2 = (Id_2, A^{c_2}, V^{c_2})$ such that $c_1 \in C_1$ and $c_2 \in C_2$.

Algorithm 1

Input: Two sets of attributes A^{c_1} and A^{c_2} from concepts c_1 and c_2

Output: The distance value $D \in [0, 1]$

Procedure:

BEGIN

1. $\overline{A^{c_2}} = A^{c_2}$

2. For each two attributes $a \in A^{c_1}$ and $b \in A^{c_2}$

 Begin

 2.1. if $a \equiv b$ then the set $\overline{A^{c_2}} = (\overline{A^{c_2}} \setminus \{b\}) \cup \{a\}$ if b does not occur in any relationships with other attributes

 2.2. if $a \equiv b$ and a is in *generalization* relationship with some attributes $m_1, \dots, m_n \in A^{c_2}$ ($n \in [1, Card(A^{c_2})]$) then the set $\overline{A^{c_2}} = \overline{A^{c_2}} \setminus \{m_1, \dots, m_n\}$ if attributes $\{m_1, \dots, m_n\}$ does not occur in any relationships with other attributes

 End

3. Generate sets $Z^{\equiv}, Z^{\uparrow}, Z^{\downarrow}$:

- 3.1. $Z^{\equiv} = A^{c_1} \cap \overline{A^{c_2}}$

- 3.2. $Z^{\uparrow} = \{(a, b) : a \in \overline{A^{c_2}} \wedge b \in A^{c_1} \wedge a \uparrow b\}$

- 3.3. $Z^{\downarrow} = \{(a, b) : a \in A^{c_1} \wedge b \in \overline{A^{c_2}} \wedge a \downarrow b\}$

4. $Card(Z) = Card(Z^{\equiv}) + Card(Z^{\uparrow}) + Card(Z^{\downarrow})$

5. Calculate weights $w^{\equiv}, w^{\uparrow}, w^{\downarrow}$:

- 5.1. $w^{\equiv} = \frac{Card(Z^{\equiv})}{Card(Z)}$

- 5.2. $w^{\uparrow} = \frac{Card(Z^{\uparrow})}{Card(Z)}$


```

5.3.  $w^\downarrow = \frac{Card(Z^\downarrow)}{Card(Z)}$ 
6.  $avg(Z^\uparrow) = \frac{\sum_{(a,b) \in Z^\uparrow} d_s(a,b)}{Card(Z^\uparrow)}$ 
7.  $d_C = w^\equiv * 0 + w^\uparrow * avg(Z^\uparrow) + w^\downarrow * 1 = w^\uparrow * avg(Z^\uparrow) + w^\downarrow$ 
8. return  $d_C$ 
END

```

The basic idea behind this algorithm is the necessity of including relationships between attributes (and eventually concepts' structures) in the process of calculating the distance between two concepts. The first step is generating the initial set of attributes from the concept c_2 . Then the algorithm modifies it by replacing all of the equivalent attributes with corresponding elements from set A_{c^1} (Step 2.1) and then simplifying it by skipping attributes that are more general (causing knowledge redundancy) in Step (2.2). In the Step 3 we generate sets describing the coverage of relationships between attributes- we determine how many attributes are equivalent, how many are more general than the others and how many are in contradiction. Cardinalities of these sets are later used to calculate weights - the idea behind this is to assure that the more relationships of certain type occur, the more they should influence the final distance value.

In Step 6 we calculate the semantic distance between attributes that are in *generalization* relationship. We utilize the distance function briefly featured in section 4.1 and described in our previous paper [11]. Then following the conditions for this function algorithm calculates the mean value - where the distance between equivalent attributes is 0, the distance between contradicting attributes is 1 and the distance between attributes in *generalization* relationship varies from 0 to 1.

Taking on that cardinalities of sets A^{c^1} and A^{c^2} are m, n we can say that the complexity of our algorithm is $O(mn)$. We can also assume that these cardinalities are almost the same, so we can estimate that the overall complexity is quadratic ($O(n^2)$). Therefore, when concepts' structures are not very large, we can say that our algorithm is effective.

Now we will give an example effects of our algorithm for the input classes from Figure 1 and attributes' semantics given in Table 1. We will present the results of comparing class structures in two different orders and also their analysis with regard to formulated postulates in Section 4.2.

First we take that concepts $c_1 = Student$ and $c_2 = Undergraduate$. Set $\overline{A^{c^2}}$ is initially equal to A^{c^2} . After applying Step 2.1, the algorithm replaces attributes *Surname* and *Promoter* with the attributes *LastName* and *Supervisor* respectively. Due to lack of redundancy Steps 2.2 does not apply any changes. Then, in Step 3, algorithm generates set describing relationship coverage of concepts' structures. Eventually we have: $Z = \{Id, LastName, Supervisor\}$, $Z^\uparrow = \{(FullName, FirstName), (FullName, LastName), (DayOfBirth, Age), (YearOfAdmission, SchoolYear), (FacultyAndDepartment, Department), (FacultyAndDepartment, Faculty)\}$ and $Z^\downarrow = \{(Suspended, IsActive)\}$. The total cardinality of set Z (calculated in Step 4) is 10. Then the algorithm calculates weights $w^\equiv, w^\uparrow, w^\downarrow$, that are equal 0.3, 0.6 and 0.1 respectively. Thanks to

simplicity of assigned attributes' semantics all distances between the ones that are in *generalization* relationship equals 0.5, so the average value is also 0.5. According to Step 7 the end distance value d_C is 0.4.

The second example takes as inputs $c_1 = \textit{Undergraduate}$ and $c_2 = \textit{Student}$. At first set $\overline{A^{c_2}} = A^{c_2}$. Then, according to Step 2.1, the algorithm replaces attributes *LastName* and *Supervisor* with attributes *Surname* and *Promoter*. Step 2.2 does not make any changes. Step 3 generates following sets: $Z = \{\textit{Id}, \textit{LastName}, \textit{Supervisor}\}$, $Z^\uparrow = \{\}$ and $Z^\downarrow = \{\textit{IsActive}, \textit{Suspended}\}$. The total cardinality of set Z is 4 and the weights calculated according to Step 5 are: $w^\equiv = 0.75$, $w^\uparrow = 0$, $w^\downarrow = 0.25$. Due to the fact that set Z^\uparrow is empty, the average distance from Step 6 is 0. Therefore, the final distance value $d_C = 0.25$, which is smaller than in the first example, so it is in line with intuition and formulated postulates.

As expected our algorithm calculating the distance function between attributes' structures does not return symmetrical results. The natural way of thinking is that simple transforming one knowledge structure into another does not imply that it is as simple the other way. This fact causes that Postulate 1 is satisfied.

It is also consistent with intuition that it is much easier to transform detailed information to more general form (consisting of less data) then the other way around. For this reason, according to the example, the distance from the concept *Student* to the concept *Undergraduate* is higher than from *Undergraduate* to *Student*. Therefore, our algorithm satisfies Postulates 2 and 3.

The calculation complexity is acceptable (it only includes comparing two sets of attributes) and when concepts' structures are not very complicated, the total time of evaluation is short.

The semantic content of concepts, their structures and attributes are also involved in the whole process - the more equivalency between attributes occurs, the closer two concepts are. This, along with the partial alignment from equation 5, provides reliable and coherent information about concepts alignment.

5 Future Works and Summary

In this paper we have presented our work on expanding ontology definitions and ontology alignment approaches with explicit semantics added to the smallest building blocks of ontologies, which are concepts' attributes. It is straightforward development of our previous work presented in [10] and [11]. We have provided effective algorithm of calculating distances between two concepts, based on formal requirements along with formulating postulates that any forthcoming modifications should meet. Illustrative example is also given to demonstrate our approach.

In the future we want to concentrate on creating experimental environment, that will provide the functionality of editing ontologies with regard to presented notions and in parallel allowing to designate valid mappings, utilizing the algorithm from section 4.3. The initial work has already been done and described in [11]. We will continue developing this system and present obtained results in upcoming publications.

In conclusion, we believe that the content of this paper can be treated as another milestone in creating standardized methodology of managing ontologies, designating mappings between them and simplifying their analysis and applications.

Acknowledgment. This research was partially supported by Polish Ministry of Science and Higher Education under grant no. 4449/B/T02/2010/39 (2010-2013).

References

1. Bellahsene, Z., Bonifati, A., Rahm, E. (eds.): Schema Matching and Mapping, 1st edn. Springer, Heidelberg (2011)
2. Euzenat, J., Shvaiko, P.: Ontology Matching, 1st edn. Springer, Heidelberg (2007)
3. Gal, A.: Enhancing the Capabilities of Attribute Correspondences. In: Schema Matching and Mapping, 2nd edn., pp. 52–73. Springer, Heidelberg (2011)
4. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: Semantic schema matching. In: Proceedings of the 10th International Conference on Cooperative Information Systems (CoopIS 2005), pp. 347–365 (2005)
5. Gruber, T.R.: A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2), 199–220 (1993)
6. Jung, J.J.: Ontology Mapping Composition for Query Transformation on Distributed Environments. *Expert Systems with Applications* 37(12), 8401–8405 (2010)
7. Magnani, M., Rizopoulos, N., Mc.Brien, P., Montesi, D.: Schema Integration Based on Uncertain Semantic Mappings. In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) ER 2005. LNCS, vol. 3716, pp. 31–46. Springer, Heidelberg (2005)
8. Marie, A., Gal, A.: Boosting Schema Matchers. In: Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part I on On the Move to Meaningful Internet Systems, pp. 283–300 (2008)
9. Nguyen, N.T.: Advanced Methods for Inconsistent Knowledge Management (Advanced Information and Knowledge Processing). Springer, Heidelberg (2008)
10. Pietranik, M., Nguyen, N.T.: Attribute Mapping as a Foundation of Ontology Alignment. In: Nguyen, N.T., Kim, C.-G., Janiak, A. (eds.) ACIIDS 2011, Part I. LNCS(LNAI), vol. 6591, pp. 455–465. Springer, Heidelberg (2011)
11. Pietranik, M., Nguyen, N.T.: Semantic Distance Measure Between Ontology Concept's Attributes. In: Proceedings of 15th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, KES 2011 (2011)
12. Staab, S., Studer, R.: Handbook on Ontologies, 2nd edn., illus., Hardcover, vol. XIX, 118, p. 121 (2009), ISBN: 978-3-540-70999-2

Author Index

- Abe, Jair Minoro I-82
Akama, Seiki I-82
Albert, František I-458
Amarowicz, Marcin II-30
Antkiewicz, Ryszard I-478
- Babič, František I-458
Bădică, Amelia II-363
Bădică, Costin II-363
Bahrammirzaee, Arash II-352
Banaszak, Zbigniew A. I-448
Barbucha, Dariusz II-322, II-332
Bartók, Juraj I-458
Będkowski, Janusz II-130, II-140
Bednár, Peter I-458
Benbouzid-Sitayeb, Fatima II-60
Biesiada, Błażej II-150
Blachnik, Marcin I-42
Bocewicz, Grzegorz I-448
Boryczka, Mariusz II-505
Boryczka, Urszula II-475, II-485
Borzemski, Leszek II-425
Bródka, Piotr I-378
Brodowski, Stanisław I-113
Buczek, Bartłomiej M. I-52
Bura, Wojciech II-505
Burdak, Robert I-123
- Candea, Ciprian II-160
Cao, Son Thanh I-254
Čapkovič, František II-110
Ceglarek, Dariusz II-40
Chao, Wei-Ming I-419
Charchalis, Adam II-261
Chmiel, Wojciech I-602
Chmielewski, Mariusz I-314
Chohra, Amine II-352
Cholewa, Wojciech II-30
Choroś, Kazimierz II-415
Chrzanowski, Paweł II-30
Chu, Shu-Chuan I-28
Ciesielczyk, Michał II-10, II-20
Czarnecki, Adam I-582
Czarnowski, Ireneusz II-301, II-322
- Dong, Yingsai I-244
Du, Yu I-244
Duda, Jerzy II-445
Dudek, Grzegorz I-468
Dudek-Dyduch, Ewa II-290
Dyk, Michał I-399
- Filipowicz, Włodzimierz II-251
Foszner, Paweł II-281
Fraś, Mariusz I-557
- Gleba, Kamil I-602
Głowacz, Andrzej I-602
Golak, Sławomir I-42
Grąbczewski, Krzysztof II-342
Granmo, Ole-Christoffer I-72
Grobelny, Piotr I-265
Grochowski, Michał I-497
Gruca, Aleksandra II-281
Grzech, Adam I-557
- Haniewicz, Konstanty II-40
Hashim, Siti Zaiton Mohd II-90
Haugland, Vegard I-72
Hera, Łukasz II-271
Hluchý, Ladislav I-458
Homenda, Władysław I-93
Huang, Hsiang-Cheh I-28
- Ilie, Sorin II-363
Indyka-Piasecka, Agnieszka I-336
- Jaffry, S. Waqar I-366
Jakóbczak, Dariusz I-173
Jaksik, Roman II-281
Janasiewicz, Tadeusz II-10, II-20
Jankowska, Beata I-546
Jankowski, Jarosław II-395
Jaworski, Bartosz II-241
Jędrzejowicz, Piotr II-301, II-311, II-322
Jeon, Hongbeom I-438
Jo, Kang-Hyun I-428

- Jung, Jason J. I-592
 Juszczyk, Przemysław II-485
 Juszczyzyn, Krzysztof I-557
- Kabziński, Jacek II-465
 Kajdanowicz, Tomasz I-378
 Kang, Su ho II-385
 Kasemthaweasab, Pornnapas I-527
 Kasprzyk, Rafał I-388
 Katarzyniak, Radosław II-70, II-120
 Kazienko, Przemysław I-378
 Kern, Rafał I-612
 Khelifati, Si Larabi II-60
 Kim, Cheol Min I-537
 Kim, Hye-Jin I-537
 Kjølleberg, Marius I-72
 Kołaczek, Grzegorz I-557
 Kolendo, Piotr II-241
 Konieczny, Lukasz I-304
 Kordos, Mirosław I-42
 Korytkowski, Marcin I-62
 Koszelew, Jolanta I-234
 Kowalczyk, Ryszard II-70
 Kozak, Jan II-475
 Kozielski, Stanisław II-271
 Koźlak, Jarosław II-100
 Kozłowski, Bartosz I-294
 Krawiec, Krzysztof I-203
 Król, Dariusz II-191
 Kucharska, Edyta II-290
 Kukła, Elżbieta I-378
 Kurutach, Werasak I-527
 Kwak, Ho-Young I-438, I-537
 Kwaśnicka, Halina I-224
 Kwiatkowski, Jan I-557
 Kwolek, Bogdan II-455
 Kwon, Young Min II-385
- Larsen, Svein-Erik I-72
 Lasota, Tadeusz I-142
 Ławrynowicz, Agnieszka I-304
 Le, My-Ha I-428
 Lee, Junghoon I-438, I-537
 Lee, Sang Joon I-537
 Lee, Seongjun I-537
 Leon, Florin II-201
 Lesiński, Wojciech I-93
 Li, Tsai-Yen I-419
 Lis, Robert A. I-152
 Lopes, Helder F.S. I-82
- Lorkiewicz, Wojciech II-70
 Luczak, Tomasz I-142
- Machová, Kristína I-356
 Madani, Kurosh II-352
 Madziar, Michał I-304
 Maleszka, Marcin II-1
 Malski, Michał I-346
 Małyśiak-Mrozek, Bożena II-271
 Marczyk, Michał II-281
 Marusak, Piotr M. I-193
 Masłowski, Andrzej II-130, II-140
 Mazurkiewicz, Jacek II-180
 Michalak, Marcin I-103
 Michalska, Katarzyna I-326, II-180
 Minaei-Bidgoli, Behrouz I-163
 Momot, Alina II-271
 Momot, Michał II-271
 Mrozek, Dariusz II-271
 Mrozek, Maciej II-191
 Muscar, Alex II-363
 Musiał, Katarzyna I-378
 Myszkowski, Paweł B. I-52
- Najgebauer, Andrzej I-388, I-478
 Nakamatsu, Kazumi I-82
 Nalepa, Grzegorz J. II-150
 Nawrocki, Mateusz I-203
 Nguyen, Linh Anh I-254, I-572
 Nguyen, Ngoc Thanh I-612, I-623, II-1
 Nowak, Aleksandra I-304
 Nowicki, Adam I-497
 Nowicki, Robert I-62
- Očevčić, Hrvoje I-517
 Ogryczak, Włodzimierz I-294
 Orłowski, Cezary I-582
 Ortiz-Arroyo, Daniel I-224
- Pacut, Andrzej II-170
 Pałka, Piotr II-80
 Pan, Jeng-Shyang I-28
 Papliński, Janusz P. I-183
 Paralič, Ján I-409, I-458
 Park, Gyung-Leen I-438, I-537
 Park, Namje I-488
 Parvin, Hamid I-163
 Pawlak, Krzysztof T. I-304
 Pawletko, Rafał II-261
 Pham, Xuan Hau I-592

- Pieczyński, Andrzej I-265
 Pierzchała, Dariusz I-388, I-399
 Pietranik, Marcin I-623
 Piskor-Ignatowicz, Cezary II-211
 Polańska, Joanna II-281
 Polański, Andrzej II-281
 Popek, Grzegorz II-70
 Potoniec, Jędrzej I-304
 Prusiewicz, Agnieszka I-557
 Putz-Leszczynska, Joanna II-170

 Qin, Zengchang I-244

 Ratajczak-Ropel, Ewa II-311, II-322
 Roddick, John F. I-28
 Rogala, Tomasz II-30
 Różewski, Przemysław II-50
 Rulka, Jarosław I-478
 Rutkowski, Leszek I-62
 Rutkowski, Wojciech II-40
 Rybiński, Henryk I-275
 Ryk, Krzysztof I-612
 Rymut, Bogusław II-455

 Sanin, Cesar I-17, I-507
 Scafeş, Mihnea II-363
 Scherer, Rafał I-62
 Selamat, Ali II-90
 Selamat, Md Hafiz II-90
 Siemiński, Andrzej II-405
 Skinderowicz, Rafał II-495
 Skorupa, Grzegorz II-120
 Skrzyński, Paweł II-435
 Śliwa, Joanna I-602
 Smętek, Magdalena I-213
 Śmierchalski, Roman II-241
 Śnieżyński, Bartłomiej II-100
 Sobocki, Janusz I-557
 Sohn, Mye II-385

 Stachowiak, Anna I-285
 Stapor, Piotr I-314
 Stawarz, Magdalena I-103
 Suchacka, Grażyna II-425
 Świątek, Paweł I-557
 Szałas, Andrzej I-254
 Szczerbicki, Edward I-17, I-507
 Szłapczyńska, Joanna II-221
 Szłapczyński, Rafał II-221, II-231
 Szuba, Tadeusz II-435
 Szwabe, Andrzej II-10, II-20
 Szwed, Piotr I-602
 Szydło, Stanisław II-435, II-445
 Szydłowski, Adam I-399
 Szymkowiak, Magdalena I-546

 Tarapata, Zbigniew I-478
 Tomaszuk, Dominik I-275
 Trawiński, Bogdan I-142, I-213
 Treur, Jan I-1, I-366
 Tutoky, Gabriel I-409

 Walkowiak, Tomasz I-326
 Wan, Tao I-244
 Wantoch-Rekowski, Roman I-478
 Wasilewski, Adam I-557
 Wieczorek, Tadeusz I-42
 Wierzbowska, Izabela II-322
 Wójcik, Robert I-448

 Yazdani, Hossein I-224

 Zachara, Marek II-211
 Żagar, Drago I-517
 Zakrzewska, Danuta I-132
 Zamfirescu, Constantin-Bala II-160
 Zatwarnicki, Krzysztof II-374
 Zhang, Haoxi I-507
 Zolfpour Arokhlo, Mortaza II-90